PENGAMANAN BASIS DATA KEUANGAN RSUD BANGKINANG MENGGUNAKAN ALGORITMA KRIPTOGRAFI RC6

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik
pada Jurusan Teknik Informatika

oleh :

SYAHRIAL RAMADHAN SIREGAR

10251020398

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU
2010

PENGAMANAN BASIS DATA KEUANGAN RSUD BANGKINANG

MENGGUNAKAN ALGORITMA KRIPTOGRAFI RC6

SYAHRIAL RAMADHAN SIREGAR

1 0 2 5 1 0 2 0 3 9 8

Tanggal Sidang : 5 Februari 2010

Periode Wisuda : Juli 2010

Jurusan Teknik Informatika

Fakultas Sains dan Teknologi

Universitas Islam Negeri Sultan Syarif Kasim Riau

ABSTRAK

RSUD Bangkinang mempunyai basis data keuangan, yang menyimpan data-data keuangan RSUD Bangkinang. Tetapi belum terdapat sebuah mekanisme pengamanan basis data, sedangkan data yang tersimpan dalam basis data harus terjamin keamanannya, karena merupakan data-data yang penting dan vital, dan seringkali menjadi target para penyerang.

Mengatasi permasalahan keamanan basis data ini, dilakukan suatu mekanisme pengamanan dengan mengimplementasikan algoritma kriptografi RC6. Penerapan kriptografi ini dilakukan dengan menambahkan modul enkripsi dan deskripsi pada aplikasi keuangan, sehingga data yang tersimpan dalam basis data merupakan data yang terenkripsi. Pengembangan sistem menggunakan bahasa pemrograman *Visual Basic* 6.0 dan basis data *Microsoft SQL Server* 2005 *Express*.

Hasil penelitian menunjukkan bahwa RC6 dapat digunakan untuk mengamankan basis data, dimana data yang terenkripsi dalam basis data keuangan dapat dideskripsi kembali dengan benar, sehingga data dapat diproses pada level aplikasi dengan baik serta tidak mengganggu struktur basis data. Dari pengujian dengan menggunakan serangan *exhaustive attack*, diperoleh kesimpulan bahwa data tidak dapat dibuka oleh pihak yang tidak berhak.

Kata Kunci: Basis Data, Keamanan, Kriptografi, RC6.

SECURITY OF MONETARY DATABASE OF BANGKINANG'S PUBLIC HOSPITAL USING CRYPTOGRAPHY RC6 ALGORITHM

SYAHRIAL RAMADHAN SIREGAR

1 0 2 5 1 0 2 0 3 9 8

Tanggal Sidang : 5th February 2010

Periode Wisuda : July 2010

Technique of Informatics Engineering Departement

Faculty of Sciences and Technology

State Islamic University of Sultan Syarif Kasim Riau

ABSTRACT

Bangkinang's Public Hospital have a monetary databases, which is keep Bangkinang's Public Hospital monetary's data. The problem is, it has no security mechanism to protect the databases. It should be guaranted and unreached from unauthorized person, because it represent vital and important data.

To overcome of security databases problems, in this final project, security mechanism has implemented using RC6 algorithm of cryptography. Cryptography is done by enhancing module of encrypt and decrypt to monetary application, so that data in databases in the form of encrypted data. The development of system uses Visual Basic 6.0 programming language and Microsoft SQL Server 2005 Express as database.

The result of this research indicate that RC6 can be used to protect databases, where encrypted data in monetary databases can be decrypted truly. So data can be processed in the application level and also does not bother databases structure. Using exhaustive attack, it found that data cannot be opened by unauthorizhed person.

Keyword : Cryptography, Database, RC6, Security.

DAFTAR ISI

Halaman

LEMBAR PERSETUJUAN ii
LEMBAR PENGESAHAN iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL iv
LEMBAR PERNYATAAN v
LEMBAR PERSEMBAHAN vi
ABSTRAK vii
ABSTRACT viii
KATA PENGANTAR ix
DAFTAR ISI xii
DAFTAR TABEL xvi
DAFTAR GAMBAR xviii
DAFTAR LAMPIRAN xx

BAB I PENDAHULUAN
1.1 Latar Belakang I-1
1.2 Rumusan Masalah I-3
1.3 Tujuan Tugas Akhir I-3
1.4 Batasan Masalah I-3
1.5 Sistematika Pembahasan I-4

BAB II LANDASAN TEORI
2.1 Kriptografi II-1
2.1.1 Defenisi Kriptografi II-1
2.1.2 Layanan Kriptografi II-3
2.1.3 Algoritma Kriptografi II-4
2.1.4 Jenis Serangan pada Kriptografi II-9
2.2 Algoritma RC6 ..... II-12

2.2.1 Pembentukan Kunci Internal II-12
2.2.1.1 Algoritma Konversi Kunci Rahasia II-13
2.2.1.2 Algoritma Inisialisasi *Array* S II-14
2.2.1.3 Algoritma Mencampurkan L dan S.....II-14
2.2.2 Proses Enkripsi dan Deskripsi II-15
2.2.2.1 Algoritma Enkripsi RC6 .....II-17
2.2.2.2 Algoritma Deskripsi RC6 .....II-21
2.3 Basis Data II-23
2.3.1 Defenisi Basis Data II-23
2.3.2 *Structured Query Language (SQL)* II-24
2.3.3 Keamanan Basis Data II-27
2.3.4 Strategi Enkripsi pada Basis Data.....II-28

BAB III METODOLOGI PENELITIAN
3.1 Tahapan Penelitian..... III-1
3.2 Studi Literatur III-2
3.3 Studi Lapangan..... III-2
3.4 Perumusan Masalah III-2
3.5 Pengumpulan Data III-3
3.6 Analisis Sistem..... III-3
3.7 Perancangan Perangkat Lunak..... III-5
3.8 Implementasi..... III-6
3.9 Pengujian Sistem..... III-6
3.10 Kesimpulan Akhir..... III-6

BAB IV ANALISIS DAN PERANCANGAN
4.1 Analisis Masalah IV-1
4.1.1 Analisis Aplikasi Keuangan RSUD Bangkinang.....IV-1
4.1.2 Basis Data Keuangan RSUD Bangkinang IV-6
4.2 Analisis Sistem Baru IV-7
4.2.1 Analisis Kebutuhan IV-12

4.2.1.1 Analisis *Input* IV-12
4.2.1.2 Analisis Proses IV-12
4.2.1.3 Analisis *Output* IV-13
4.2.2 Analisis Basis Data pada Sistem Baru IV-13
4.2.2.1 Analisis Tabel yang Perlu Dienkripsi dan Tidak Perlu Dienkripsi IV-13
4.2.2.2 Perubahan Struktur Tabel_BKU IV-14
4.2.3 Analisis Fungsional IV-16
4.2.3.1 *Context Diagram* Aplikasi Keuangan IV-16
4.2.3.2 Perubahan pada DFD Level 3 Pengelolaan Data Akses IV-17
4.2.3.3 Perubahan pada DFD Level 3 Pengelolaan Data Transaksi IV-18
4.2.3.4 Perubahan pada DFD Level 3 Pengelolaan Data Transaksi Buku Bank IV-20
4.2.4 Analisis Data IV-21
4.3 Analisis Penerapan Algoritma RC6 dalam Enkripsi Basis Data IV-27
4.3.1 Algoritma Pembangkit Sub Kunci IV-30
4.3.2 Algoritma Baca Masukan untuk Proses Enkripsi IV-31
4.3.3 Algoritma *Whitening* Awal IV-32
4.3.4 Algoritma Iterasi IV-32
4.3.5 Algoritma *Whitening* Akhir IV-33
4.3.6 Algoritma Baca Masukan untuk Proses Deskripsi IV-34
4.3.7 Algoritma Deskripsi IV-35
4.3.8 Perhitungan Manual Algoritma RC6 ..... IV-36
4.4 Perancangan IV-41
4.4.1 Perancangan Basis Data IV-41
4.4.2 Perancangan Modul Perangkat Lunak IV-42
4.4.3 Perancangan Antarmuka IV-43

BAB IV IMPLEMENTASI DAN PENGUJIAN

5.1 Lingkungan Implementasi V-1
5.2 Batasan Implementasi V-2
5.3 Implementasi Modul Perangkat Lunak V-2
5.4 Implementasi Antarmuka V-2
5.4.1 Antarmuka Masukan Data V-3
5.4.2 Antarmuka Enkrip Data V-3
5.4.3 Antarmuka Dekrip Data V-5
5.5 Pengujian V-8
5.5.1 Pengujian Enkripsi dan Deskripsi V-8
5.5.2 Pengujian Keamanan Data Terenkripsi V-10
5.5.3 Kesimpulan Pengujian V-12

BAB VI KESIMPULAN DAN SARAN

6.1. Kesimpulan VI-1
6.2. Saran VI-1

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR RIWAYAT HIDUP

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sistem manajemen basis data adalah suatu kumpulan dari data yang saling terhubung dan suatu program yang dapat mengakses data tersebut (Silberschatz,2002). Kumpulan data tersebut kemudian lebih dikenal dengan istilah basis data. Basis data mengandung informasi yang sesuai dengan kebutuhan organisasi yang menggunakannya. Tujuan utama dari sistem manajemen basis data yaitu untuk menyediakan jalan untuk menyimpan dan mendapatkan kembali informasi pada basis data dengan nyaman dan efisien.

Data yang tersimpan dalam basis data harus terjamin kemanannya, karena merupakan data-data yang penting dan vital dan seringkali menjadi target bagi para penyerang. Tidak adanya mekanisme pengamanan dapat menyebabkan data-data ini diketahui dan dirusak oleh pihak yang tidak berhak. Bentuk ancaman yang dilakukan oleh penyerang dapat berupa ancaman pasif (*passive attack*), yaitu dengan sengaja mengambil, membaca dan menampilkan data, dan ancaman aktif (*active attack*), yaitu memodifikasi bahkan memalsukan data yang tersimpan dalam basis data (Meyer,1982).

Rumah Sakit Umum Daerah (RSUD) Bangkinang telah mempunyai aplikasi keuangan, yaitu Aplikasi Laporan Pertanggung Jawaban Bendahara Pengeluaran dan terhubung dengan basis data keuangan, yang digunakan untuk menyimpan, mengelola dan membuat laporan keuangannya tepat waktu, akurat

dan efisien. Basis data keuangan ini belum mempunyai sebuah mekanisme pengamanan, sehingga permasalahan keamanan basis data ini menjadi penting untuk mencegah pihak yang tidak berhak mengetahui merusak data-data keuangan tersebut.

Untuk mengatasi permasalahan keamanan basis data keuangan ini, maka perlu dikembangkan sebuah mekanisme pengamanan pada aplikasi tersebut. Pengamanan dapat dilakukan melalui dua cara. Cara pertama ialah pengaturan hak akses setiap pengguna oleh administrator basis data. Cara yang kedua ialah pengamanan dari sisi kandungan data yang tersimpan pada basis data. Pada cara pertama masih ditemukan kelemahan, karena kebocoran data dapat disebabkan oleh pihak yang tidak berhak, yaitu pihak dalam selain pemilik data, dalam hal ini pengguna aplikasi keuangan.

Maka dalam penulisan tugas akhir ini dipilih cara kedua, salah satunya dengan cara mengimplementasikan kriptografi. Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan (Schenier,1999). Algoritma kriptografi adalah aturan untuk melakukan proses enkripsi yaitu proses menyandikan dari plainteks menjadi cipherteks dan proses dekripsi yang merupakan kebalikan dari enkripsi.

Penerapan kriptografi ini dilakukan dengan mengamankan sisi substansi (kandungan data) yang tersimpan pada basis data. Secara teknis, penerapan kriptografi ini dilakukan dengan mengubah data asli (*plaintext*) menjadi data tersandi (*chipertext*). Data yang diolah diambil dari basis data dan data hasil pengolahan disimpan *(update*) kembali pada basis data yang bersangkutan sehingga data yang tersimpan pada basis data merupakan data tersandi.

Algoritma RC6 adalah algoritma blok chiper yang akan dipergunakan dalam penulisan laporan tugas akhir ini. RC6 dipilih karena algoritma ini merupakan algoritma yang sangat aman, padat dan sederhana dan menawarkan performansi yang sangat bagus dan fleksibel (Rivest, 1998). Algoritma ini memiliki performansi yang sangat baik walaupun besar chiperteks selalu sedikit lebih besar daripada besar plainteks, memiliki *avelache effect* yang baik, dan tidak memiliki kunci lemah ataupun kunci setengah lemah. Sampai saat ini, belum ada serangan yang secara signifikan dapat memecahkan kunci dari algoritma RC6 dalam tempo waktu yang singkat.

Diharapkan dengan mengimplementasikan algoritma kriptografi RC6 dapat meningkatkan keamanan basis data dari ancaman seperti tersebut diatas tanpa mengurangi performa basis data secara signifikan.

1.2 Rumusan Masalah

Permasalahan yang akan diselesaikan dalam tugas akhir ini adalah bagaimana melakukan pengamanan basis data keuangan, dengan mengimplementasikan algoritma RC6 pada aplikasi keuangan.

1.3 Tujuan Tugas Akhir

Tujuan penulisan tugas akhir ini adalah menerapkan algoritma RC6 untuk keamanan data pada aplikasi keuangan RSUD Bangkinang.

1.4 Batasan Masalah

Dalam pelaksanaan tugas akhir ini ditetapkan beberapa batasan yang akan dijadikan pedoman dalam pelaksanaan tugas akhir.

1. Enkripsi hanya dilakukan pada kolom-kolom dari tabel yang menyimpan transaksi keuangan, yaitu tabel_bku dan tabel_buku_bank, serta tabel yang menyimpan data akses, yaitu tabel akses.
2. Algoritma RC6 yang akan digunakan menggunakan panjang blok sebesar 32 bit, jumlah putaran sebanyak 20 putaran dan panjang kunci 16 *byte*.
3. Parameter uji yang akan digunakan untuk pengujian implementasi RC6 adalah *Exhaustive Key Search*, sebagai salah satu serangan yang paling baik terhadap algoritma RC6.

1.5 Sistematika Pembahasan

Sistematika pembahasan dalam penulisan laporan tugas akhir ini terdiri dari pokok-pokok permasalahan yang dibahas pada masing-masing bab yang diuraikan menjadi beberapa bagian sebagai berikut :

1. Bab I Pendahuluan, menjelaskan hal-hal yang menjadi latar belakang pelaksanaan tugas akhir, perumusan masalah, penentuan tujuan, batasan masalah dan sistematika pembahasan dalam laporan tugas akhir.
2. Bab II Landasan Teori, menguraikan teori-teori yang berkaitan dan mendukung pelaksanaan tugas akhir.
3. Bab III Metodologi Penelitian, menguraikan metodologi yang digunakan dalam pelaksanaan tugas akhir.
4. BAB IV Analisis dan Perancangan, menguraikan bagian analisis yang terdiri dari analisis masalah, analisis sistem baru dan analisis algoritma RC6. Selain itu juga menguraikan bagian perancangan basis data, perancangan modul perangkat lunak dan perancangan antarmuka.

5. BAB V Implementasi dan Pengujian, menguraikan berisi uraian hasil implementasi dan pengujian terhadap perangkat lunak yang dibuat pada tugas akhir ini.
6. BAB VI Kesimpulan dan Saran, menguraikan kesimpulan dan saran dari tugas akhir ini secara keseluruhan.

BAB II

LANDASAN TEORI

Bab kedua ini berisi penjelasan tentang dasar teori yang berkaitan dan mendukung dalam pelaksanaan tugas akhir ini. Dalam bab ini akan dibahas gambaran umum kriptografi berikut algoritma-algoritma kriptografi khususnya algoritma RC6. Selain itu juga terdapat pembahasan tentang basis data berikut pembahasan bahasa *SQL* (*Structured Query Language*), keamanan basis data dan strategi enkripsi pada basis data.

2.1 Kriptografi

Berikut ini akan dijelaskan tentang definisi kriptografi, layanan-layanan kriptografi, algoritma kriptografi dan jenis-jenis serangan pada kriptografi.

2.1.1 Definisi Kriptografi

Kriptografi memiliki berbagai macam pengertian, secara etimologis kata kriptografi berasal dari bahasa Yunani yang terdiri atas dua kata yaitu κρυπτο (baca : krupto) yang berarti tersembunyi atau rahasia dan γραφη (baca: grafh) yang berarti tulisan. Karena itu kriptografi dapat diartikan sebagai tulisan rahasia, sedangkan menurut istilah ada beberapa pengertian, diantaranya yaitu :

1. Kriptografi adalah ilmu sekaligus seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya.

2. Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan (Silberschatz,2002).
3. Kriptografi adalah cara dan ilmu untuk mengkodekan (enkripsi dan dekripsi) data sehingga data tersebut hanya dapat didekodekan oleh orang tertentu.

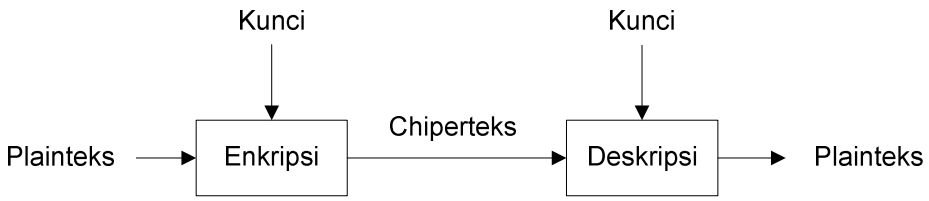
Sistem kriptografi (*Cryptosystem*) adalah sebuah sistem yang terdiri atas algoritma kriptografi yang berfungsi untuk mengacak data plainteks dengan satu atau beberapa kunci yang berupa angka atau *string* yang hanya diketahui oleh pihak pengirim dan penerima. Hasil akhir dari proses ini berupa cipherteks.

Suatu sistem kriptografi yang kuat memiliki kemungkinan jangkauan kunci yang sangat besar sehingga sistem ini tidak mungkin dipecahkan dengan mencoba semua kemungkinan kunci secara *brute force*. Sistem kriptografi yang kuat juga akan menciptakan cipherteks yang acak untuk semua standar tes statistik (Fauzan, 2008).

Pesan adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Ada dua jenis pesan yaitu plainteks dan cipherteks. Plainteks adalah pesan yang dapat langsung dibaca dan dimengerti artinya, sedangkan cipherteks adalah pesan yang telah disandikan sehingga tidak bermakna lagi. Cipherteks bertujuan agar pesan tidak dapat dibaca oleh pihak yang tidak berhak.

Dalam kriptografi terdapat proses enkripsi (*encryption*) yaitu proses menyandikan plainteks menjadi cipherteks atau disebut juga *enciphering*. Proses lain yaitu dekripsi (*decryption*) yaitu proses mengembalikan cipherteks menjadi plainteks semula atau disebut juga *deciphering*.

Berikut ilustrasi proses enkripsi dan dekripsi dapat dilihat pada gambar II-1 (Munir,2006).



Gambar 2-1 Proses Enkripsi dan Dekripsi

Secara matematis enkripsi dapat digambarkan sebagai :

C = cipherteks

P = plainteks

Fungsi enkripsi E memetakan P ke C,

$$E(P) = C$$

Fungsi dekripsi D memetakan C ke P,

$$D(C) = P$$

Fungsi enkripsi dan dekripsi harus memenuhi sifat :

$$D(E(P)) = P$$

2.1.2 Layanan Kriptografi

Kriptografi menyediakan beberapa layanan, yaitu (Munir, 2006) :

1. Kerahasiaan (*confidentiality*)

   Layanan yang digunakan untuk menjaga isi pesan dari siapapun yang tidak berhak untuk membacanya.

2. Integritas data (*data integrity*)

   Layanan yang menjamin bahwa pesan masih asli/utuh atau belum pernah dimanipulasi selama pengiriman.

3. Otentikasi (*authentication*)

   Layanan untuk mengidentifikasi kebenaran pihak-pihak yang berkomunikasi (*user authentication*) dan mengidentifikasi kebenaran sumber pesan (*data origin authentication*).

4. Nirpenyangkalan (*non-repudiation*)

   Layanan untuk mencegah entitas yang berkomunikasi melakukan penyangkalan, yaitu pengirim pesan menyangkal melakukan pengiriman atau penerima pesan menyangkal telah menerima pesan.

2.1.3 Algoritma Kriptografi

Algoritma kriptografi adalah aturan untuk *enciphering* dan *deciphering* atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi pesan (Munir, 2006). Algoritma kriptografi berkembang terus dan terbagi atas dua bagian yaitu algoritma kriptografi klasik dan algoritma kriptografi modern. Pada algoritma klasik, kriptografer menggunakan algoritma sederhana, yang memungkinkan cipherteks dapat dipecahkan dengan mudah. Pemecahan algoritma klasik seperti penggunaan statistik kemunculan huruf pada bahasa tertentu, terkaan, intuisi dan sebagainya, sedangkan algoritma kriptografi modern dibuat sedemikian kompleks sehingga kriptanalis sangat sulit memecahkan cipherteks tanpa mengetahui kunci. Ciri khas umum algoritma modern yaitu beroperasi dalam mode *bit*.

Berdasarkan jenis kunci yang digunakan, algoritma kriptografi modern dapat dibedakan menjadi dua kategori yaitu :

1. Algoritma kriptografi simetris.

Algoritma kriptografi simetris yaitu algoritma yang menggunakan hanya satu kunci untuk enkripsi dan dekripsi.

Contoh algoritma simetris yaitu DES (*Data Encryption Standard*), Rijndael, *Blowfish, IDEA, GOST, Serpent, RC2, RC4, RC5,* dan lain-lain.

Skema algoritma simetris dapat dilihat pada gambar II-2 (Munir, 2006).

Gambar 2-2 Algoritma Simetris

2. Algoritma kriptografi nir-simetris atau algoritma kunci publik.

   Algoritma kriptografi nir-simetris yaitu algoritma yang menggunakan kunci publik untuk proses enkripsi dan privat untuk proses dekripsi.

   Contoh algoritma Nir-Simetris yaitu RSA.

   Skema algoritma nir-simetris dapat dilihat pada gambar II-3 (Munir, 2006).

Gambar 2-3 Algoritma Nir-Simetris

Algoritma kriptografi simetri dapat dikelompokkan menjadi dua kategori yaitu :

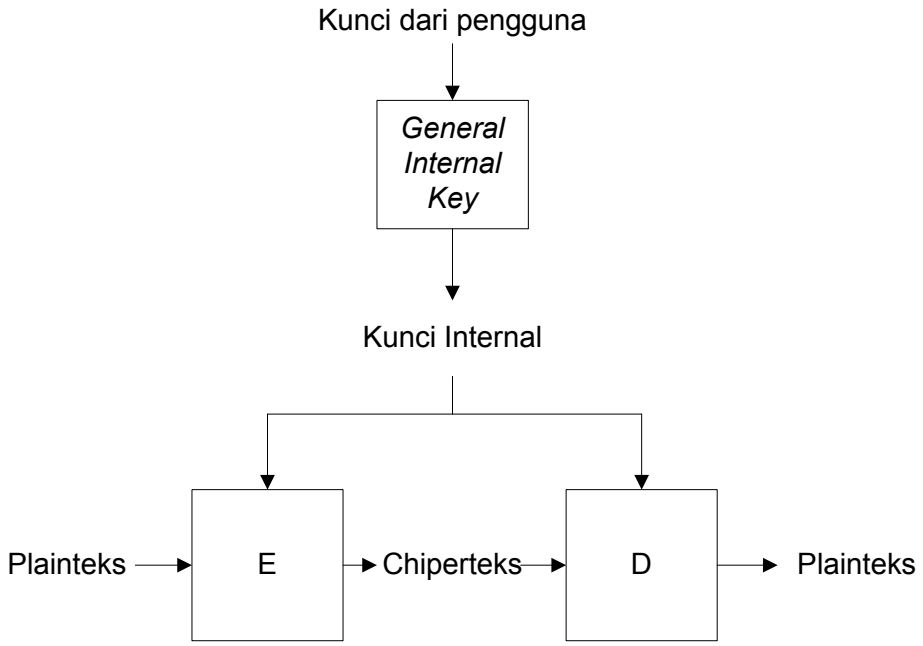
1. Cipher Aliran (*stream cipher*)

   Cipher aliran adalah algoritma kriptografi yang beroperasi pada plainteks atau cipherteks dalam bentuk *bit* tunggal, yang dalam hal ini rangkaian *bit* dienkripsikan atau didekripsikan *bit* per*bit* (Munir,2006).

   Contoh *stream cipher* adalah *RC4, Seal, A5, Oryx,* dan lain-lain

2. Cipher blok (*block cipher*)

   Block cipher adalah suatu tipe algoritma kriptografi kunci simetris yang mengubah plainteks yang dibagi dalam blok-blok dengan panjang yang sama menjadi cipherteks yang memiliki panjang blok yang sama. Ukuran panjang blok dapat beragam bergantung kepada algoritma yang digunakan, ukuran yang sering digunakan adalah 64 bit dan menuju 128 bit. Seperti semua algoritma kunci simetri, proses enkripsi yang dilakukan akan menggunakan suatu *input* dari *user* yang disebut sebagai kunci rahasia. Kunci rahasia ini juga akan dipakai ketika melakukan proses dekripsi. Cara kerja secara umum dari *block cipher* dapat dilihat pada Gambar II-2 (Munir, 2006).



Gambar 2-4 Skema Cara Kerja *Block Cipher*

Dalam melakukan perancangan *block cipher,* beberapa prinsip harus dipertimbangkan. Prinsip-prinsip tersebut yaitu (Munir,2006) :

1. Prinsip *Confusion* dan *Diffusion* dari Shannon.

   Tujuan dari prinsip *confusion* adalah untuk menyembunyikan hubungan apapun yang ada antara plainteks, cipherteks, dan kunci, sehingga dapat membuat kriptanalis kesulitan dalam menemukan pola-pola pada cipherteks.

   Tujuan dari prinsip *diffusion* adalah menyebarkan pengaruh satu bit plainteks atau kunci ke sebanyak mungkin cipherteks, sehingga dengan berubahnya satu bit plainteks dapat mengubah cipherteks yang sulit untuk diprediksi.

2. *Iterated Cipher*

   Untuk menambah keamanan, pada algoritma-algoritma *block cipher* dilakukan iterasi pada pemrosesan setiap blok, pada setiap rotasi dari iterasi tersebut digunakan fungsi transformasi yang sama namun memakai kunci yang berbeda yang disebut dengan kunci internal. Kunci internal pada umumnya merupakan hasil dari kunci yang dimasukan oleh pengguna yang dikomputasi menggunakan suatu fungsi tertentu. Dengan adanya iterasi tersebut keamanan akan semakin terjamin, namun performansi akan berkurang karena adanya waktu lebih yang dibutuhkan untuk melakukan iterasi. *Block cipher* yang menerapkan konsep iterasi ini disebut juga dengan *iterated block cipher.*

3. Kunci Lemah

   Suatu hal yang perlu dihindari dalam melakukan perancangan algoritma kriptografi adalah kunci lemah, yaitu jika sebuah plainteks dienkripsi ganda menggunakan kunci tersebut akan menghasilkan plainteks itu sendiri.

Hal lain yang perlu diperhatikan dalam perancangan *blok chiper* adalah *padding*, yaitu penambahan blok terakhir dengan pola bit teratur agar panjang blok terakhir sama dengan blok yang ditetapkan (Munir, 2006), karena terdapat kemungkinan panjang plainteks tidak habis dibagi dengan panjang ukuran blok yang ditetapkan, misalnya 32 bit dan sebagainya. Hal

ini mengakibatkan blok terakhir berukuran lebih pendek daripada blok lainnya.

Misalkan ukuran blok adalah 32 bit, dan blok terakhir terdiri dari 24 bit. Maka blok terakhir ditambahkan dengan 8 bit, misalnya dengan menambahkan 8 buah bit 0. Setelah proses deskripsi, 8 bit terakhir dari blok deskripsi terakhir dihapus untuk mendapatkan plainteks kembali.

2.1.4 Jenis Serangan pada Kriptografi

Dalam algoritma kriptografi ada beberapa jenis serangan yang terdefinisi, antara lain (Rudianto,2008):

1. *Exhaustive Key Search*

Penyerang mencoba semua kemungkinan kunci satu persatu dan mengecek apakah plainteks memiliki kecocokan dengan cipherteks yang menjadi sampel. Untuk sebuah blok cipher dengan k-bit kunci dan n-bit blok, jumlah pasangan cipherteks dan plainteks pengujian yang diperlukan untuk menentukan kunci yang tepat adalah sekitar k/n. Tambahan, jika ada plainteks yang sifatnya redundan, serangan hanya dapat bekerja dengan baik hanya jika sebagian blok cipherteks diketahui. Jumlah blok cipherteks yang dibutuhkan bergantung pada frekuensi kata-kata yang sering muncul.

2. *The matching cipherteks attack*

Serangan ini didasarkan pada fakta bahwa setidaknya ada sebuah blok cipher berukuran m-bit yang muncul dari hasil enkripsi yang berasal dari 2m/2 blok plainteks sehingga dapat diketahui sedikit informasi mengenai plainteksnya.

3. *Differential crytanalysis*

Cara ini merupakan salah satu metode kriptanalisis konvensional yang paling umum dan sering digunakan, yang dipublikasikan oleh Bilham dan Shamir pada tahun 1990. Kriptanalisis ini biasa digunakan untuk melawan metode-metode kriptografi yang dibangun dari perulangan fungsi yang tetap.

Salah satu caranya adalah dengan mencari selisih antara dua (2) bit string, misalnya X dan X' persamaan (1) $\Delta X = X \Phi (X')-1$ dimana Φ adalah kumpulan operasi terhadap kumpulan bit string yang digunakan untuk mengombinasikan kunci dengan plainteks dalam fungsi putaran dan dimana (X) -1 adalah *inverse* elemen dari X. Ide di balik metode ini adalah selisih dari plainteks dan cipherteks, yang didapatkan dari hasil kombinasi dengan kunci, selalu sama besarnya.

4. *Truncated differentials*

Untuk beberapa cipherteks, dimungkinkan dan sangat bermanfaat, memprediksi hanya sebagian nilai saja dengan menggunakan *differential crytanalysis* untuk setiap putarannya

5. *Impossible differentials*

Salah satu tipe dari *differential crytanalysis* yang kemungkinannya adalah 0. Ide utamanya adalah menspesifikkan bahwa ketidak-mungkinan terhadap beberapa putaran terhadap cipher serangan. Kemudian dengan menebak beberapa kunci dalam putaran yang tidak tercakup dalam fungsi, dapat dilakukan pembuangan terhadap beberapa nilai kunci yang salah.

6. *Higher-order differentials*

Sebuah *sth-order differentials* didefinisikan secara rekursif sebagai sebuah fungsi *differentials* dari fungsi (*s-1)th-order differentials*, dimana *sth-order differentials* berisi kumpulan 2n teks yang mengandung *predetermined differentials*.

7. *Linear crytanalysis*

Linear crytanalysis ditemukan oleh Matsui pada tahun 1993. *Linear crytanalysis* merupakan serangan *knownplainteks* dimana penyerang mengeksploitasi pendekatan persamaan linier dari beberapa bit plainteks, beberapa bit cipherteks, dan beberapa bit kunci.

8. Kriptanalisis Mod n

Serangan ini merupakan generalisasi dari serangan linier. Serangan ini dapat digunakan untuk cipherteks dimana beberapa kata terbiaskan dalam persamaan modulo n, dimana n adalah integer yang bernilai kecil. Telah terbukti bahwa algoritma kriptografi yang menggunakan hanya rotasi bit dan menambahkan modulo dari 2^{32} sangat rapuh terhadap serangan ini.

9. *Related-key attacks*

Knudsen telah menggunakan metode mendapatkan hasil enkripsi menggunakan satu kunci terhadap sebuah plainteks yang terpilih dan berhasil mengurangi kunci secara *exhaustive search* sampai dengan empat (4) kali lebih cepat. Serangan ini membutuhkan hasil enkripsi dari beberapa kunci yang berbeda, dalam beberapa kasus juga memerlukan plainteks yang sama, oleh karena itu, cara ini dianggap kurang realistik.

2.2 Algoritma RC6

Algoritma RC6 adalah salah satu algoritma kriptografi *block cipher*, dirancang oleh Ronald L. Rivest, Matt J.B. Robshaw, Ray Sidney, dan Yuqin Lisa Yin dari RSA Laboratories. Algoritma RC6 ini berhasil menjadi finalis dan menjadi kandidat kuat untuk menjadi AES (*Advanced Encryption Standard*). Versi 1.1 dari RC6 mulai dipublikasikan pada tahun 1998. Dasar desain dari algoritma RC6 ini didasarkan pada pendahulunya yaitu algoritma RC5.

Algoritma RC6 merupakan algoritma dengan parameter penuh, algoritma RC6 dispesifikasikan dengan notasi RC6-*w/r/b.* Dimana *w* adalah ukuran dari *word* dalam bit, karena pada RC6 menggunakan 4 buah register maka *word* adalah ukuran blok dibagi 4. Paramater *r* menunjukkan banyaknya iterasi selama proses enkripsi dan deskripsi, dimana bilangan *r* tidak boleh negatif. Dan *b* adalah panjang kunci dalam *bytes*. Setelah algoritma ini masuk ke AES, maka ditetapkan panjang blok sebesar 128 bit, sehingga ukuran masing-masing register adalah panjang blok dibagi 4, sehingga ukuran *w* adalah 32 bit, jumlah iterasi *r* sebesar 20 kali putaran, dan *b* bervariasi antara 16, 24 dan 32 *byte*.

Cara kerja dari algoritma RC6 adalah menggunakan 4 buah register dan menggunakan prinsip *Iterated Block Cipher* yang mengunakan iterasi.

2.2.1 Pembentukan Kunci Internal

Untuk membangkitkan urutan kunci internal yang akan digunakan selama proses enkripsi, algoritma RC6 melakukan proses pembangunan kunci bertujuan untuk membangun suatu *array* S yang berukuran 2*r*+4 dari kunci masukan

pengguna sepanjang *b bytes* ($0 \leq b \leq 255$), *array* tersebut akan digunakan baik dalam proses enkripsi maupun dekripsi.

Proses untuk membangun kunci-kunci internal menggunakan dua buah konstanta yang disebut dengan *magic constant*. Berikut nilai *magic constant* pada panjang blok 32 bit dalam heksadesimal:

P_{32} = *b7e15163*

Q_{32} = *9e3779b9*

Langkah-langkah pembangunan kunci terdiri dari tiga tahap, dan akan dijelaskan dengan algoritma-algoritma berikut ini :

2.2.1.1 Algoritma Konversi Kunci Rahasia dari *bytes* ke *words*

Algoritma Konversi Kunci Rahasia dari *bytes* ke *words* adalah sebagai berikut (Rivest, 1998) :

```
if b=0 then
      c←1
endif
for i ←b-1 downto 0 do
      L[i/u] ← (L[i/u]<<<8) + K[i]
endfor
```

Langkah pertama adalah menyalin kunci rahasia K[0..b-1] kedalam sebuah *array* L[0..c-1], dimana c = pembulatan keatas (*b*/u) dan u = *w*/8, penyalinan tersebut dilakukan secara *little endian.* Untuk semua posisi *byte* pada L yang kosong diberi nilai nol. Untuk kasus dimana *b* = 0, maka *c* = 1 dan L[0] = 0.

2.2.1.2 Algoritma Inisialisasi *Array* S

Algoritma inisialisasi *array* S adalah sebagai berikut (Rivest, 1998) :

```
S[0] ← P_w
for i ←0 to 2r+3 do
      S[i] ← S[i]+ Q_w
endfor
```

Algoritma diatas adalah untuk melakukan inisialisasi *array* S agar memiliki pola *pseudo*-random bit tertentu menggunakan progresi aritmatika modulo 2^w yang ditentukan dengan P_w dan Q_w.

2.2.1.3 Algoritma Mencampurkan L dan S

Algoritma Mencampurkan L dan S (Rivest, 1998) adalah sebagai berikut :

```
i←0
j←0
A←0
B←0
V←3*max(c,2r+4)
for index←1 to v do
      S[i]←(S[i]+A+B) <<< 3
      A←S[i]
      L[j]←(L[j]+A+B) <<< (A+B)
      B←L[j]
      i←(i+1)mod(2r+4)
      j←(j+1)mod c
endfor
```

mpurkan kunci rahasia dari pengguna yang sudah tersimpan dalam L dengan S sebanyak 3 kali iterasi.

Pembentukan kunci yang dilakukan, mengubah kunci dari *user* yang panjangnya beragam (0-255) menjadi suatu rangkaian kunci dengan sepanjang

word sebanyak 2*r+3* buah. Hal ini menjadikan RC6 dapat bekerja dengan kunci masukan pengguna yang beragam.

Kunci yang dihasilkan oleh proses pembentukan kunci ini memiliki sifat satu arah (Rivest, 1998), sehingga proses pembentukan kunci ini dapat digunakan sebagai fungsi *hash* satu arah. Dengan sifat satu arah tersebut, maka kunci internal akan sangat berbeda dengan kunci yang dimasukkan oleh pengguna, hal ini akan membuat hubungan statistik antara kunci yang dimasukan oleh pengguna dengan plainteks dan cipherteks menjadi lebih rumit karena dalam melakukan enkripsi, kunci yang dipakai adalah kunci internal.

Pada pembentukan kunci internal digunakan iterasi yang cukup banyak baik pada tahap satu, dimana untuk melakukan ekspansi kunci dibutuhkan iterasi, dan pada tahap dua, dimana dibutuhkan iterasi untuk melakukan inisialisasi *array* serta pada tahap terakhir yang dibutuhkan untuk menggabungkan dua buah *array*, yang bahkan dilakukan selama tiga kali. Iterasi-iterasi ini membutuhkan waktu yang cukup besar untuk dilakukan.

2.2.2 Proses Enkripsi dan Dekripsi

Algoritma RC6 bekerja dengan empat buah register A,B,C,D yang masing-masing berukuran *w*-bit, register-register tersebut akan diisi oleh plainteks yang kemudian akan digunakan selama proses enkripsi dan setelah proses enkripsi berakhir isi dari register-register tersebut merupakan cipherteks.

Byte pertama dari plainteks atau cipherteks akan disimpan pada *least significant byte* dari A dan *byte* terakhir dari plainteks atau cipherteks disimpan

pada *most significant byte* dari D. Proses enkripsi dan dekripsi algoritma RC6 menggunakan enam buah operasi dasar:

$a + b$ = penjumlahan integer modulo 2^w

$a - b$ = pengurangan integer modulo 2^w

$a \oplus b$ = operasi *bitwise exclusive-or* sebesar *w*-bit *words*

$a * b$ = perkalian integer modulo 2^w

$a<<<b$ = rotasi sejumlah *w*-bit *word* ke kiri sebanyak jumlah yang diberikan oleh *least sifnificant* lg *w* bit dari b

$a>>>b$ = rotasi sejumlah *w*-bit *word* ke kanan sebanyak jumlah yang diberikan oleh *least sifnificant* lg *w* bit dari b

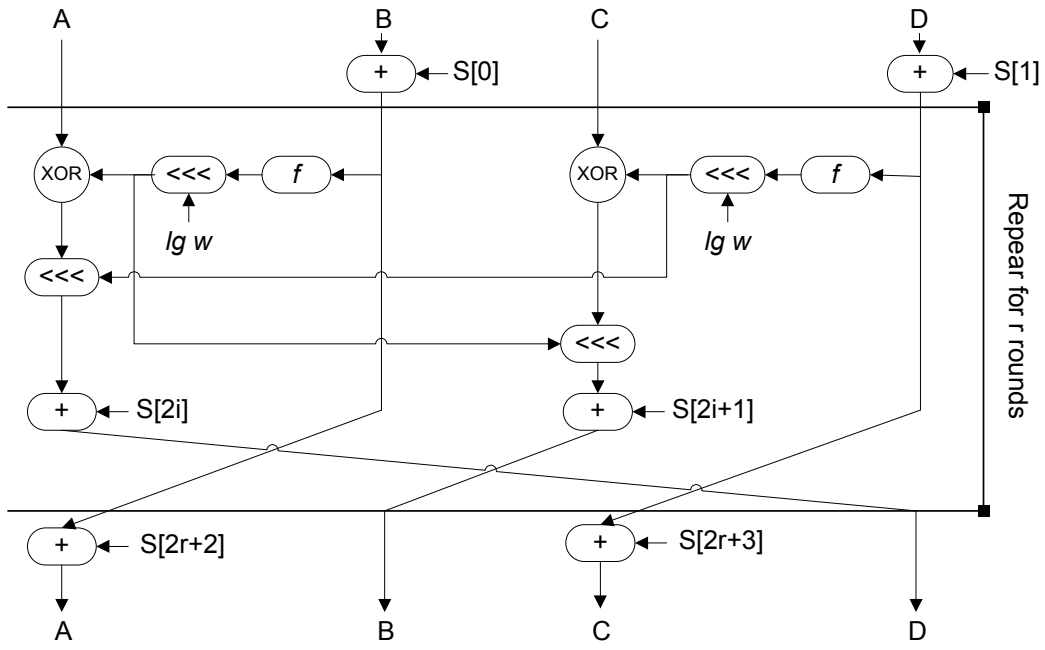
Dimana lg *w* adalah logaritma basis dua dari *w*.

Proses enkripsi dan deskripsi dapat dilihat pada algoritma berikut ini, dengan $f(x)=x * (2x+1)$ (Rivest,1998).

2.2.2.1 Algoritma Enkripsi RC6

Algoritma Enkripsi RC6 adalah sebagai berikut (Rivest, 1998) :

```
Prosedure Enkripsi ( Input      : Plainteks dalam A,B,C,D
                     r          : integer (jumlah rotasi)
                     S[0..2r+3] : kunci internal
                     Output     : Chiperteks dalam A,B,C,D)
Kamus
      u : integer
      t : integer
Algoritma
      B ← B + S[0]
      D ← D + S[1]
      for i ← 1 to r do
            t ← (B*(2B+1))<<<lg w
            u ← (D*(2D+1))<<<lg w
            A ← ((A ⊕ t)<<<u) + S[2i]
            C ← ((C ⊕ u)<<<t) + S[2i+1]
            (A, B, C, D) ← (B, C, D, A)
      endfor
      A ← A + S[2r + 2]
      C ← C + S[2r + 3]
```



Gambar 2-5 Diagram Enkripsi RC6

Langkah-langkah enkripsi algoritma RC6 secara detil adalah sebagai berikut :

1. Blok plainteks dibagi menjadi 4 bagian, A, B, C dan D yang masing-masing memiliki panjang *w* bit atau panjang blok dibagi 4. Kemudian B dan D dijumlahkan (dalam modulo 2^w) dengan kunci internal S[0] dan S[1].

```
B ← B + S[0]
D ← D + S[1]
```

2. Selanjutnya pada setiap putaran dari 1 sampai *r*, lakukan XOR dan pergeseran kekiri terhadap A dengan *f(x)* yang di geser ke kiri sebanyak lg *w*, dimana *f(x)= x* (2x+1)* dan *x* = B. Setelah itu melakukan penjumlahan (dalam modulo 2^w) dengan kunci internal. Hal serupa dilakukan pula terhadap C dengan *x* = D. Kemudian melakukan *swapping* A ← B, B ← C, C ← D dan D ← A.

```
for i ← 1 to r do
      t ← (B*(2B+1))<<<lg w
      u ← (D*(2D+1))<<<lg w
      A ← ((A ⊕ t)<<<u) + S[2i]
      C ← ((C ⊕ u)<<<t) + S[2i+1]
      (A, B, C, D) ← (B, C, D, A)
endfor
```

Fungsi *f(x) = x * (2x+1)* memiliki keistimewaan dalam diterapkan pada *iterated cipher*, keistimewaannya adalah fungsi ini memiliki sifat satu ke satu pada aritmatik modulo 2^w dan cenderung merubah bit yang

high-order (dekat MSB) (Rivest, 1998). Sifat satu arah tersebut dapat terlihat sebagai berikut :

Misalkan *A* dan *B* adalah bilangan bulat positif dan $A \neq B$, jika $A * (2A + 1) = B * (2B + 1) \pmod{2^w}$, maka:

$\Leftrightarrow \quad 2A^2 + A = 2B^2 + B \pmod{2^w}$

$\Leftrightarrow \quad 2A^2 - 2B^2 + A - B = 0 \pmod{2^w}$

$\Leftrightarrow (A - B)(2A + 2B + 1) = 0 \pmod{2^w}$

Namun, $A \neq B$, jadi $(A - B) \neq 0$ kemudian, 2*A* dan 2*B* merupakan genap sehingga (2*A* + 2*B* + 1) merupakan bilangan ganjil dan tidak mungkin nol, maka tidak ada *A* dan *B* yang memenuhi $A * (2A + 1) = B * (2B + 1) \pmod{2^w}$ atau *f(x)* bersifat satu ke satu pada modulo 2^w .

Sifat satu ke satu ini cenderung berbeda pada bit yang *high-order* atau menuju MSB, hal ini dikarenakan fungsi $f(x) = x * (2x + 1)$ merupakan fungsi kuadratik dimana pada perkalian dua buah bilangan akan cenderung menambah digit didepan. Apabila *x* pada *f(x)* yang terdiri dari *i* bit mengalami perubahan bit pada posisi ke *j* maka *f(x)* akan berubah pada bit posisi ke *j* dan cenderung pada posisi > *j*.

Dengan sifat satu ke satu pada fungsi *f(x)* tersebut, maka kemungkinan hasil *f(x)* yang berulang dalam iterasi-iterasi yang terjadi akan sangat kecil, sehingga semakin banyak jumlah iterasi, maka keamanan akan semakin terjaga, hal ini diperkuat dengan kemungkinan perubahan yang terjadi pada *high-order bit* sehingga pengaruh perbuatan lebih besar.

Jika tidak terdapat sifat satu ke satu tersebut, algoritma enkripsi akan menjadi tidak baik, karena pada algoritma ini terdapat XOR dan apabila suatu bilangan di XOR-kan 2 kali maka bilangan tersebut akan muncul kembali.

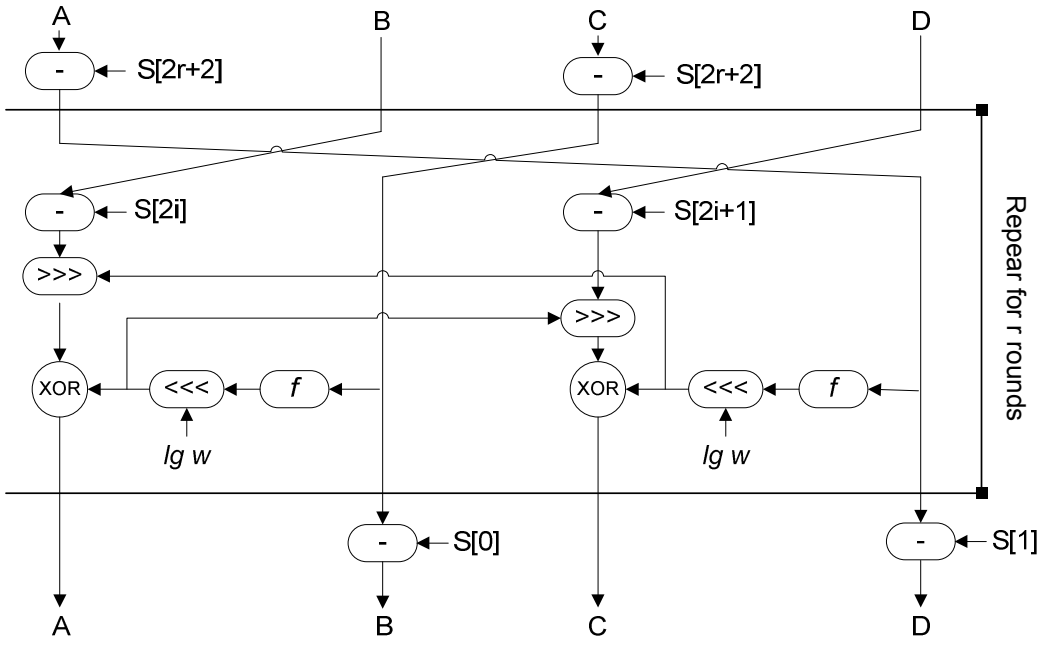
3. Setelah iterasi selesai langkah terakhir adalah melakukan penjumlahan (dalam modulo 2^w) terhadap A dan C dengan dua kunci internal terakhir. Setelah semua selesai blok yang terbagi menjadi 4 bagian disatukan kembali.

```
A ← A + S[2r + 2]
C ← C + S[2r + 3]
```

2.2.2.2 Algoritma Deskripsi RC6

Algoritma Dekripsi RC6 adalah sebagai berikut (Rivest, 1998) :

```
Prosedure Deskripsi ( Input      : Chiperteks dalam A,B,C,D
                      r          : integer (jumlah rotasi)
                      S[0..2r+3] : kunci internal
                      Output     : Plainteks dalam A,B,C,D)
Kamus
      u : integer
      t : integer
Algoritma
      C ← C - S[2r + 3]
      A ← A - S[2r + 2]
      for i ← r downto 1 do
            (A, B, C, D) ← (D, A, B, C)
            u ← (D*(2D+1))<<<lg w
            t ← (B*(2B+1))<<<lg w
            C ← ((C - S[2i+1])>>>t)⊕ u
            A ← ((A - S[2i])>>>u)⊕ t
      endfor
```



Gambar 2-6 Diagram Dekripsi RC6

Algoritma RC6 termasuk kedalam *iterated cipher*, kekuatan utama algoritma ini terletak pada iterasi yang dilakukannya. Dengan dilakukannya iterasi yang berulang-ulang dengan menggunakan kunci yang berbeda-beda, maka prinsip *confusion* dan *diffusion* dilakukan secara berulang-ulang pula, sehingga keamanan akan semakin baik.

RC6 juga memperlihatkan sebuah *avallanche effect* yang baik, yaitu 50% dari besar blok penyandian. *Avellanche effect* adalah perubahan yang kecil pada plainteks maupun *key* akan menyebabkan perubahan yang signifikan terhadap

cipherteks yang dihasilkan. Suatu *avallanche effect* dikatakan baik jika perubahan bit yang dihasilkan berkisar antara 45-60% (sekitar separuhnya, 50 % adalah hasil yang sangat baik) (Rivest, 1998). Hal ini dikarenakan perubahan tersebut berarti membuat perbedaan yang cukup sulit untuk kriptanalis melakukan serangan.

Serangan yang paling baik untuk memecahkan algoritma RC6 adalah serangan dengan menggunakan *exhaustive key search* yang ditujukan kepada kunci yang dimasukkan oleh pengguna atau kunci internal. Untuk serangan yang lebih rumit seperti kriptanalisis diferensial dan linier, dapat digunakan untuk memecahkan algoritma RC6 yang menggunakan jumlah rotasi yang kecil, untuk jumlah rotasi 20 keatas, serangan ini tidak dapat bekerja dengan baik karena sulitnya menemukan karakteristik iteratif yang baik atau perkiraan linier (Rivest, 1998).

2.3 Basis Data

Berikut akan dijelaskan tentang definisi basis data, *Structured Query Language (SQL)* dan keamanan basis data.

2.3.1 Definisi Basis Data

Secara etimologi basis data terdiri dari dua kata yaitu basis dan data yang dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, atau kombinasinya.

Basis data dapat didefinisikan dalam sejumlah sudut pandang seperti :

1. Himpunan kelompok data (arsip) yang saling berhubungan dan diorganisasi sedemikian rupa sehingga dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan *file* / tabel / arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

Sistem basis data adalah suatu sistem yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan lainnya dan membuatnya dapat digunakan untuk beberapa aplikasi.

2.3.2 *Structured Query Language* (SQL)

Suatu basis data mempunyai bahasa khusus yang diperlukan untuk melakukan interaksi dengan basis data itu sendiri. Bahasa basis data yang menjadi standar adalah *SQL* (bahasa *query* yang terstruktur). Basis data menyediakan *Data Definition Language* (*DDL*) yang menspesifikasikan skema basis data dan *Data Manipulation Language* (*DML*) yang mengekspresikan *query* basis data dan meng-*update* basis data. Pada praktiknya *DDL* dan *DML* bukanlah dua bagian yang terpisah, tetapi *DDL* dan *DML* itu merupakan suatu bagian bentuk sederhana dari suatu basis data.

1. *Data Definition Language* (*DDL*)

DDL digunakan untuk menspesifikasikan skema basis data, antara lain : membuat tabel baru, membuat indeks, mengubah struktur tabel dan sebagainya. Berikut contoh pernyataan dalam bahasa *SQL* untuk mendefinisikan tabel data_pelanggan

```
create table data_pelanggan (no_pelanggan integer, nama
char(20), alamat char(20))
```

Selain menspesifikasikan relasi, *DDL* juga menspesifikasikan informasi dari tiap relasi, seperti domain tipe, *primary key*, dan batasan lain sehingga sistem basis data harus selalu mengecek pada batasan tersebut sewaktu terjadi penambahan atau pengubahan data.

SQL standar mendukung berbagai domain tipe, diantaranya yaitu *char*(n), *varchar*(n), *int*, *smallint*, *numeric*(p,d), *real*, *float*(n), *date*, *time* dan *timestamp*.

2. *Data Manipulation Language*(*DML*)

Manipulasi data adalah :

a. Mengambil informasi yang tersimpan dalam basis data.
b. Menyisipkan informasi baru ke dalam basis data.
c. Menghapus informasi dari basis data.
d. Mengubah informasi yang tersimpan dalam basis data.

DML adalah suatu bahasa yang dapat digunakan untuk mengakses dan memanipulasi data yang terorganisir oleh suatu model data (Silberschatz,2002). *DML* terbagi atas dua tipe, yaitu :

1. *DML* prosedural, dimana pengguna harus menspesifikasikan data apa yang dibutuhkan dan bagaimana mendapatkan data tersebut

2. *DML* deklaratif atau Non-prosedural *DML*, dimana pengguna hanya menspesifikasikan data yang dibutuhkan tanpa menspesifikasikan bagaimana cara mendapatkan data itu. *DML* jenis ini adalah *DML* yang secara umum dikenal, contohnya adalah *SQL language.*

Query adalah suatu pernyataan permintaan untuk mengambil suatu informasi. Bagian dari DML yang dapat digunakan untuk mengambil informasi disebut *query language.* Meskipun secara teknis kata *query language* tidak sama dengan *DML* tetapi pada praktiknya kedua istilah ini sama.

Berikut beberapa contoh penggunaan *query languages* :

```
select nama from data_pelanggan where no_pelanggan = 13
```

Query di atas berfungsi untuk menampilkan data dari tabel data_pelanggan dengan atribut no_pelanggan = 13.

```
insert into data_pelanggan (no_pelanggan, nama, alamat)
values (13, “Mr.X”, “Jalan Ganesha No. 1000”)
```

Query di atas berfungsi untuk menyisipkan data baru ke tabel data_pelanggan. Kolom yang akan ditambah datanya adalah kolom no_pelanggan, nama, alamat dengan nilai no_pelanggan = 13 nama = “Mr.X” dan alamat “Jalan Ganesha No. 1000”.

```
update data_pelanggan set alamat = “Jalan Taman Sari No.
50/56” where no_pelanggan = 13
```

Query di atas berfungsi untuk melakukan *update* atribut alamat untuk baris pada tabel data_pelanggan dengan no_pelanggan = 13.

```
delete from data_pelanggan where no_pelanggan = 13
```

Query di atas berfungsi untuk menghapus data pada baris yang mengandung atribut no_pelanggan = 13.

2.3.3 Keamanan Basis Data

Jenis kejahatan pada basis data yaitu (Silberschatz,2002) : pembacaan informasi, pemodifikasian dan perusakan data oleh orang yang tidak memiliki otoritas. Keamanan basis data berarti menjaga basis data dari ancaman tersebut.

Persoalan keamanan pada basis data dapat dikategorikan menjadi beberapa level, yaitu (Silberschatz,2002) :

1. Sistem Basis Data

   Sistem basis data yang digunakan harus dapat menjamin setiap pengguna basis data tidak melanggar otoritas yang dimiliki masing-masing pengguna. Pengguna basis data hanya dapat memakai basis data sesuai dengan wewenang yang dimiliki dan diatur oleh administrator basis data.

2. Sistem Operasi

   Apabila tingkat keamanan pada sistem basis data telah terjamin, hal lain yang perlu diperhatikan adalah keamanan pada sistem operasi yang digunakan. Sistem operasi yang tidak aman dapat menyebabkan pengguna yang tidak memiliki otoritas ke basis data dapat mengakses basis data.

3. Jaringan

   Pada umumnya, suatu sistem basis data digunakan secara luas melalui jaringan. Keamanan jaringan yang dipakai oleh sistem basis data menjadi

hal yang penting untuk diperhatikan. Data yang ditransmisikan dari *server* basis data dengan *client* harus aman dari pihak yang tidak terotentikasi.

4. Fisik

   Keamanan level ini menyangkut keamanan yang berkaitan dengan tempat dimana sistem basis data berada. Tempat tersebut harus dilindungi dari ancaman secara fisik, seperti dirusak pencuri atau bencana alam.

5. Manusia

Setiap pengguna basis data harus diatur otoritasnya sedemikian rupa sehingga setiap pengguna hanya dapat mengakses data yang berhak diakses oleh pengguna yang bersangkutan

2.3.4 Strategi Enkripsi pada Basis Data

Berdasarkan perkembangan teknologi pengamanan saat ini, terdapat dua strategi alternatif yang dapat digunakan dalam pengamanan basis data menggunakan enkripsi, yaitu dengan enkripsi secara internal dalam basis data dengan memanfaatkan fitur yang telah didukung oleh DBMS dan dengan melakukan enkripsi secara eksternal di luar basis data.

1. Enkripsi secara internal dalam basis data

Strategi ini merupakan strategi pengamanan yang paling sederhana karena dilakukan hanya dengan memanfaatkan fitur enkripsi yang telah digunakan oleh DBMS yang bersangkutan atau dengan menggunakan produk *add-on* yang menambahkan fitur enkripsi pada DBMS yang belum memiliki kemampuan

tersebut. Enkripsi internal basis data dapat dilakukan dengan *whole database encryption* atau dengan *column encryption*.

Strategi enkripsi basis data secara internal memiliki beberapa kelemahan. Proses enkripsi dan dekripsi akan menambah beban proses yang harus dijalankan oleh sistem sehingga performansi DBMS akan menurun cukup tajam. Degradasi performansi ini akan lebih terasa jika digunakan *whole database encryption.* Selanjutnya, data yang perlu diproteksi masih rawan diserang saat berpindah dari satu sistem ke sistem lain karena di luar basis data data tersebut berada dalam bentuk plainteks. Penanganan lebih lanjut akan diperlukan untuk mengamankan transfer data di luar basis data.

Dengan enkripsi data secara internal, pada umumnya kunci akan disimpan di dalam tabel di dalam basis data yang sama. Artinya, data yang diproteksi dengan enkripsi tidak terpisah dengan kunci yang digunakan untuk mengenkripsi dan mendekripsinya. Meskipun kunci umumnya disimpan dalam tabel dengan akses terbatas, hal ini tentu saja akan meningkatkan resiko keamanan data karena setiap orang yang dapat mengakses basis data juga dapat mengakses kunci.

2. Enkripsi secara eksternal di luar basis data

Strategi penyimpanan data yang lebih aman adalah dengan menambahkan fungsi enkripsi pada aplikasi. Enkripsi dilakukan di dalam aplikasi sehingga data dapat ditransfer dan disimpan dalam bentuk terenkripsi. Pendekatan ini menyediakan pengamanan *end-to-end* yang baik, namun membutuhkan perubahan pada aplikasi yaitu dengan menambahkan atau memodifikasi fungsi enkripsi dan dekripsi.

Salah satu langkah efektif untuk mengimplementasikan strategi ini adalah dengan membangun server enkripsi yang menyediakan layanan enkripsi secara terpusat (*centralized encryption service*) untuk seluruh *environment* basis data. Cara ini dapat menyederhanakan proses manajemen dan meningkatkan kontrol terhadap *environment* multi-aplikasi yang menggunakan banyak basis data. Server enkripsi dapat dioptimalkan untuk menjalankan operasi kriptografi yang diminta oleh aplikasi. Server menjadi basis fungsi enkripsi yang dapat dipanggil oleh setiap aplikasi pada sistem.

Kelebihan utama dari pengimplementasian strategi ini adalah memberikan sistem pengamanan kunci yang terbaik. Data yang telah dienkripsi dimasukan ke dalam basis data, sedangkan kuncinya tetap berada pada server enkripsi. Hal ini meningkatkan proteksi pada basis data.

Pengimplementasian strategi ini tentunya membutuhkan sistem pengamanan yang ketat terhadap aplikasi dan server enkripsi. Solusinya adalah dengan menerapkan sistem otentikasi sehingga hanya *user* yang memiliki otoritas saja yang dapat mendekripsi data sensitif dengan mengakses kunci yang disimpan dalam server enkripsi. Solusi kedua adalah dengan meningkatkan sensitifitas dari server enkripsi dengan melakukan *monitoring* terhadap aktivitas user yang mencurigakan dan mengaudit log kejadian secara reguler.

Kelebihan lain yang didapat dari strategi ini adalah peningkatan performansi karena server basis data (DBMS) tidak dibebani dengan pemrosesan kriptografi (fungsi enkripsi). Strategi ini juga memungkinkan kemampuan untuk menambah fungsi enkripsi sesuai kebutuhan. Pembangunan server enkripsi dan modifikasi

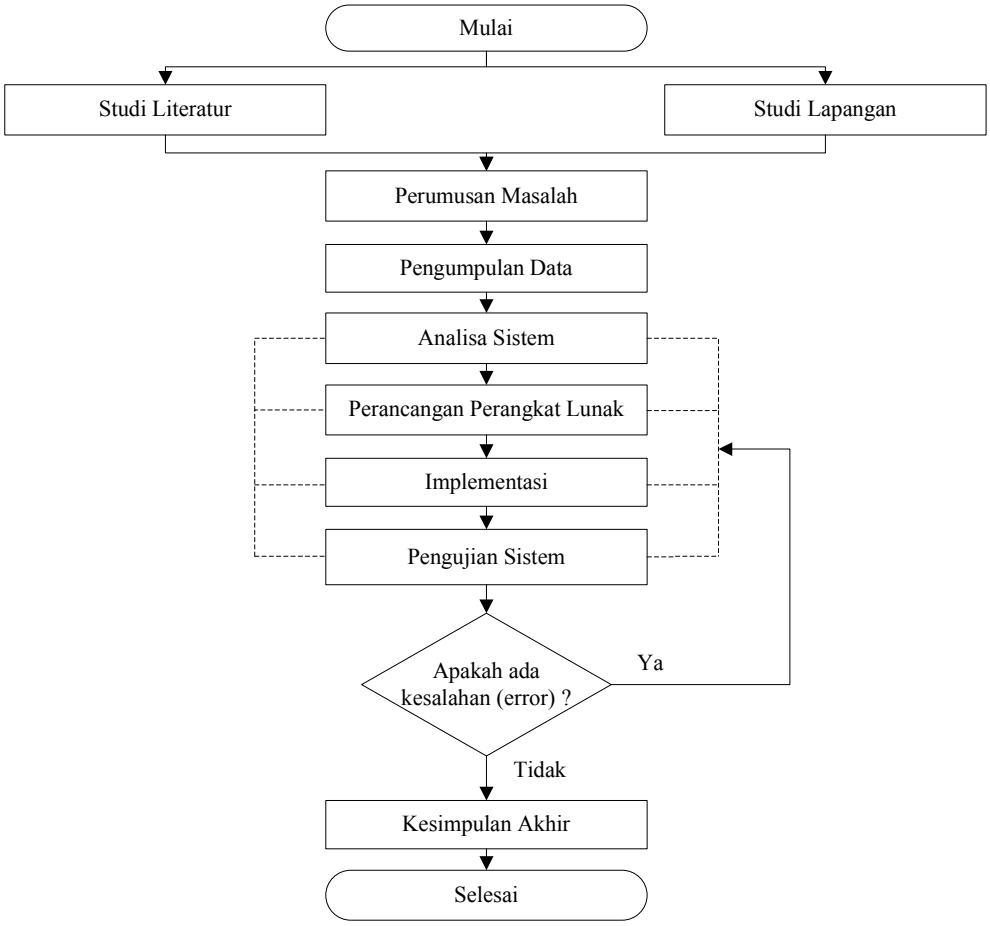
aplikasi merupakan kerja yang berat dan juga membutuhkan biaya yang cukup besar. Namun, strategi ini memberikan sistem pengamanan dan performansi yang lebih baik (Hapsari, 2005).

BAB III

METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Metodologi penelitian digunakan sebagai pedoman dalam pelaksanaan penelitian agar hasil yang dicapai tidak menyimpang dari tujuan yang telah dilakukan sebelumnya. Tahap-tahap yang akan dilalui dalam metodologi penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Tahapan Penelitian

3.2 Studi Literatur

Pada tahap ini dilakukan studi terhadap beberapa alat bantu dan konsep yang akan digunakan dalam pembuatan tugas akhir ini. Studi dilakukan pada beberapa alat bantu yang akan digunakan untuk membangun sistem dalam tugas akhir ini seperti *Microsoft Visual Basic* 6.0 dan *Microsoft SQL Server 2005 Express*.

Studi juga dilakukan dengan mempelajari berbagai macam buku teks, diktat kuliah, jurnal, karya tulis ilmiah, tugas akhir dan tesis yang berkaitan dengan masalah yang akan dibahas yaitu kriptografi khususnya algoritma kriptografi RC6, basis data dan sistem manajemen basis data, sehingga penulis mendapatkan dasar-dasar referensi yang kuat dalam menentukan metode yang tepat untuk menyelesaikan permasalahan yang akan diteliti.

3.3 Studi Lapangan

Pada tahap ini dilakukan studi pada RSUD Bangkinang untuk dapat mengetahui permasalahan-permasalahan yang ada, yang kemudian akan dirumuskan saat melakukan perumusan masalah.

3.4 Perumusan Masalah

Pada tahap ini ditentukan masalah yang akan diselesaikan dalam tugas akhir ini, yaitu pengamanan basis data keuangan RSUD Bangkinang, dengan mengimplementasikan algoritma RC6 pada aplikasi keuangan.

3.5 Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data tentang aplikasi dan basis data keuangan yang disebutkan diatas. Semua tahap pada proses pengumpulan data-data tersebut diperoleh dari wawancara dan observasi.

a. Wawancara (*interview*)

   Proses wawancara dilakukan dengan pihak-pihak yang berhubungan dengan aplikasi dan pembuat program, untuk mendapatkan informasi tentang aplikasi dan struktur basis data yang ada**.**

b. Pengamatan (*observasi*)

   Observasi merupakan salah satu teknik pengumpulan data yang efektif untuk mempelajari suatu sistem. Hal ini dilakukan dengan pengamatan secara langsung terhadap aplikasi keuangan serta pengamatan langsung terhadap basis datanya.

3.6 Analisis Sistem

Setelah melakukan pengumpulan data, langkah berikutnya adalah menganalisis sistem yang akan dibuat sesuai dengan batasan yang ada. Dalam tahap ini, terdapat beberapa langkah yang dilakukan antara lain :

1. Analisis Masalah

   Langkah yang dilakukan pada tahap ini adalah menganalisis permasalahan atau kelemahan yang terdapat pada sistem lama yang sedang berjalan, yaitu aplikasi keuangan RSUD Bangkinang. Pada

tahap ini diketahui bahwa belum terdapat suatu mekanisme pengamanan pada basis data keuangan tersebut

2. Analisis Sistem Baru

   Pada tahap ini akan ditentukan kebutuhan sistem yang baru sebagai solusi dari kekurangan sistem yang telah ada, yaitu dengan dengan menambahkan modul pada aplikasi yang mendukung proses enkripsi dan deskripsi sehingga data yang tersimpan berupa data tersandi

   Pada tahap analisis sistem baru akan dilakukan analisis kebutuhan, analisis tabel yang akan dienkripsi, analisis struktur tabel, analisis data dan analisis fungsional.

   Analisis kebutuhan meliputi analisis *input,* proses dan *output.* Pada analisis *input* akan dibahas *field- field* apa yang menjadi masukan untuk proses *enchipering* dan *dechipering*.

   Sedangkan analisis proses berupa :

   a. Ketika dilakukan perintah *insert* untuk menyimpan data ke tabel-tabel yang dimaksud diatas, data terlebih dahulu dienkrip agar data yang tersimpan dalam tabel berupa chiperteks.
   b. Ketika dilakukan perintah *select* dari tabel-tabel diatas, dilakukan deskripsi untuk mengembalikan data menjadi plainteks, sehingga bisa diolah pada aplikasi.

   Output dari sistem ini adalah data yang tersimpan pada basis data berupa chiperteks.

Pada analisis tabel yang dienkripsi, akan ditentukan tabel mana yang perlu dienkripsi dan tabel mana yang tidak perlu dienkripsi.

Dalam melakukan analisis data akan menggunakan alat bantu ER-Diagram dan dalam analisis fungsional akan menggunakan alat bantu berupa *Data Flow Diagram* (DFD).

3. Analisis Algoritma

   Dalam tahap ini dilakukan analisis penerapan algoritma yang akan dipergunakan yaitu algoritma RC6 dalam melakukan enkripsi dan deskripsi basis data keuangan, berdasarkan analisa sistem lama dan sistem baru yang akan dibangun. Analisis ini meliputi langkah demi langkah yang dilakukan terhadap data yang dienkripsi dan dideskripsi, serta meliputi perhitungan manual algoritma RC6 dalam melakukan enkripsi dan deskripsi data.

3.7 Perancangan Perangkat Lunak

Pada tahap ini dilakukan perancangan sesuai hasil analisis sistem, khususnya perancangan basis data, perancangan modul-modul enkripsi dan deskripsi, dan modul pendukung lainnya yang akan diintegrasikan dengan aplikasi, serta perancangan antarmuka.

Dalam pengembangan perangkat lunak akan digunakan metode konvensional dengan menggunakan model *Waterfall*. Model ini mensyaratkan penyelesaian suatu tahap secara tuntas sebelum beranjak pada tahap sebelumnya dan hasil masing-masing tahap harus didokumentasikan secara baik.

3.8 Implementasi

Pada proses implementasi ini dilakukan pembuatan modul-modul yang telah dirancang dalam tahap perancangan kedalam bahasa pemrograman tertentu. Dalam hal ini aplikasi ini akan menggunakan :

a. Perangkat lunak yang digunakan dalam pembuatan dan penerapan aplikasi menggunakan *Microsoft Visual Basic* 6.0 dan *database* menggunakan *Microsoft SQL Server* 2005 *Express*.
b. Perangkat keras yang digunakan dalam pembuatan dan penerapan aplikasi adalah:
    1. *Processor* Intel Pentium 1,50 GHz
    2. *Memory* 512 MB
    3. *Harddisk* berkapasitas 80 GB

    Monitor, Mouse dan Keyboard

3.9 Pengujian Sistem

Tahap pengujian dilakukan dengan tujuan untuk menjamin sistem yang dibuat sesuai dengan hasil analisis dan perancangan serta menghasilkan satu kesimpulan apakah sistem tersebut sesuai dengan yang diharapkan. Untuk itu dibutuhkan sebuah metode pengujian yang menjadi ukuran atau paramater sehingga dapat ditarik kesimpulan bahwa sistem memang telah berjalan sesuai dengan tujuan. Metode pengujian yang digunakan adalah salah satu dari serangan-serangan pada kriptografi, yaitu metode *exhaustive search*.

3.10 Kesimpulan Akhir

Pada tahap ini diambil kesimpulan akhir dalam penerapan algoritma RC6 pada basis data keuangan, berdasarkan hasil pengujian yang telah dilakukan, untuk mengetahui apakah implementasi algoritma RC6 yang telah dilakukan telah dapat melakukan pengamanan terhadap basis data dengan baik. Pada tahap ini juga diberikan saran untuk perbaikan pengembangan sistem ini.

BAB IV

ANALISIS DAN PERANCANGAN

Pada bab ini akan dilakukan analisis yang bertujuan untuk mencari solusi dari permasalahan pengamanan basis data keuangan, menguraikan bagian analisis yang terdiri analisis masalah, analisis sistem baru serta analisis algoritma RC6 dalam enkripsi dan deskripsi basis data. Selain itu juga menguraikan bagian perancangan basis data, perancangan modul perangkat lunak dan perancangan antarmuka.

4.1 Analisis Masalah

Langkah yang dilakukan pada tahap ini adalah menganalisis permasalahan berdasarkan perumusan masalah. Pada subbab berikut akan dibahas analisis dari aplikasi keuangan dan basis data keuangan RSUD Bangkinang.

4.1.1 Analisis Aplikasi Keuangan RSUD Bangkinang

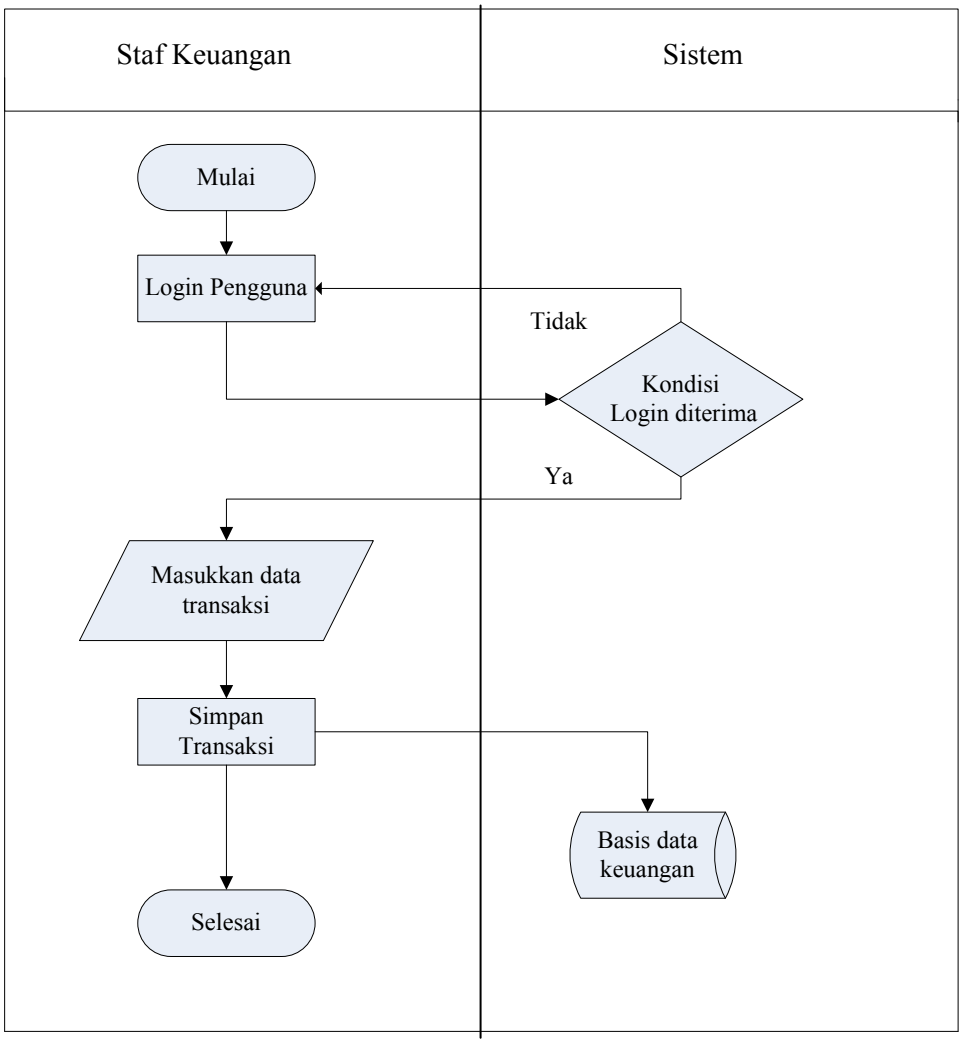
Aplikasi Keuangan RSUD Bangkinang, yang diberi nama Laporan Pertanggung Jawaban Bendahara Pengeluaran RSUD Bangkinang, mempunyai fungsi untuk mengelola data keuangan, khususnya di bagian Bendaharawan Pengeluaran. Data-data penting dan vital yang menyangkut data-data keuangan RSUD Bangkinang ini disimpan pada sebuah basis data yang diberi nama dbKeuangan. Aplikasi ini menggunakan pengamanan berupa otentikasi pengguna untuk dapat menjalankan aplikasi.

Kelemahan pada aplikasi ini adalah belum terdapat suatu mekanisme pengamanan untuk mengamankan basis data, sehingga data-data penting tersebut dapat diketahui oleh pihak yang tidak berhak.

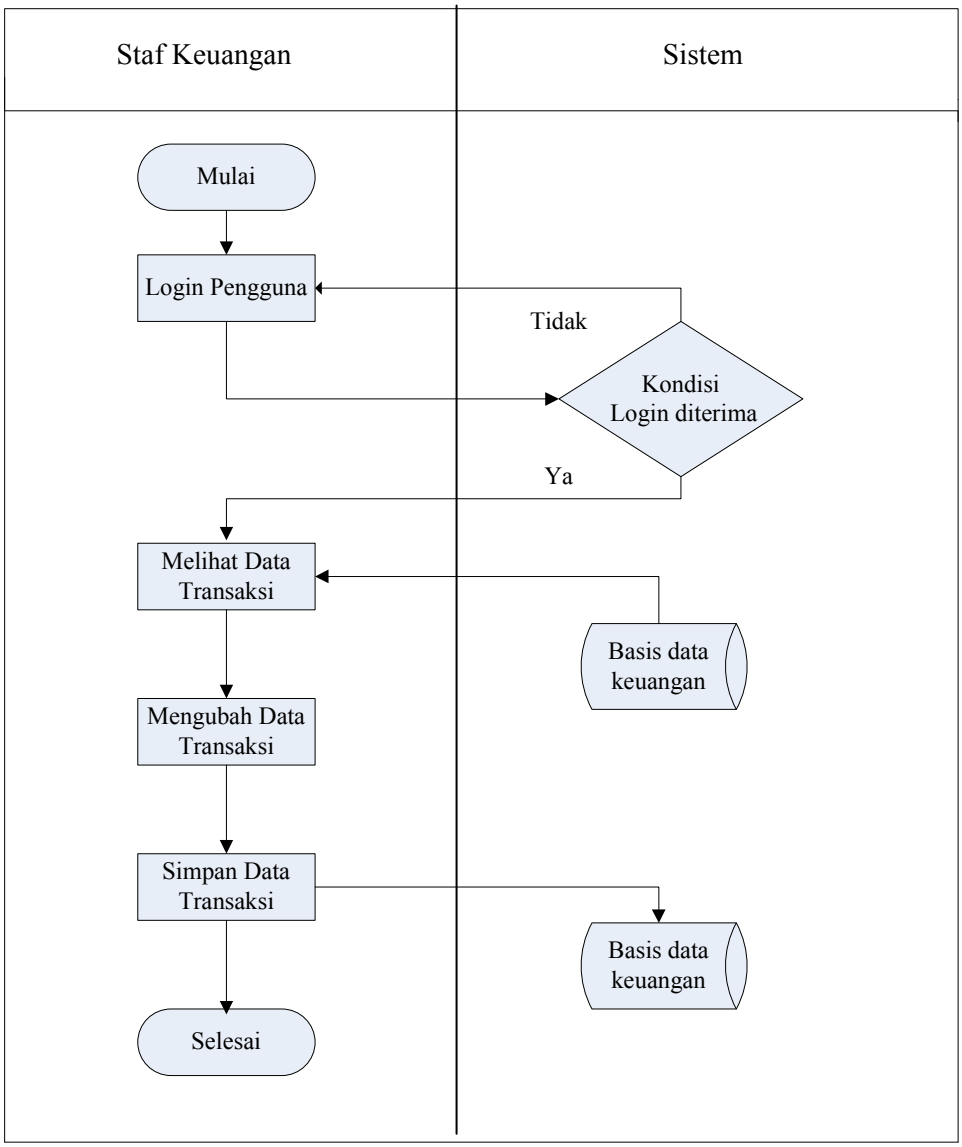
Pengguna dalam aplikasi ini adalah Bagian Keuangan, yang berhak untuk mengelola hak akses pengguna, pengaturan aplikasi dan pengaturan anggaran, yang meliputi pengaturan belanja, pengaturan program, pengaturan kegiatan, pengaturan kode rekening dan jumlah anggaran. Serta berhak mengelola data transaksi keuangan, meliputi pengelolaan buku kas umum, rincian obyek, buku pajak, buku rekening bank dan laporan.

Aplikasi ini dikembangkan dengan menggunakan bahasa pemrograman *Microsoft Visual Basic* 6.0, dan menggunakan basis data *Microsoft SQL Server* 2005 *Express*.

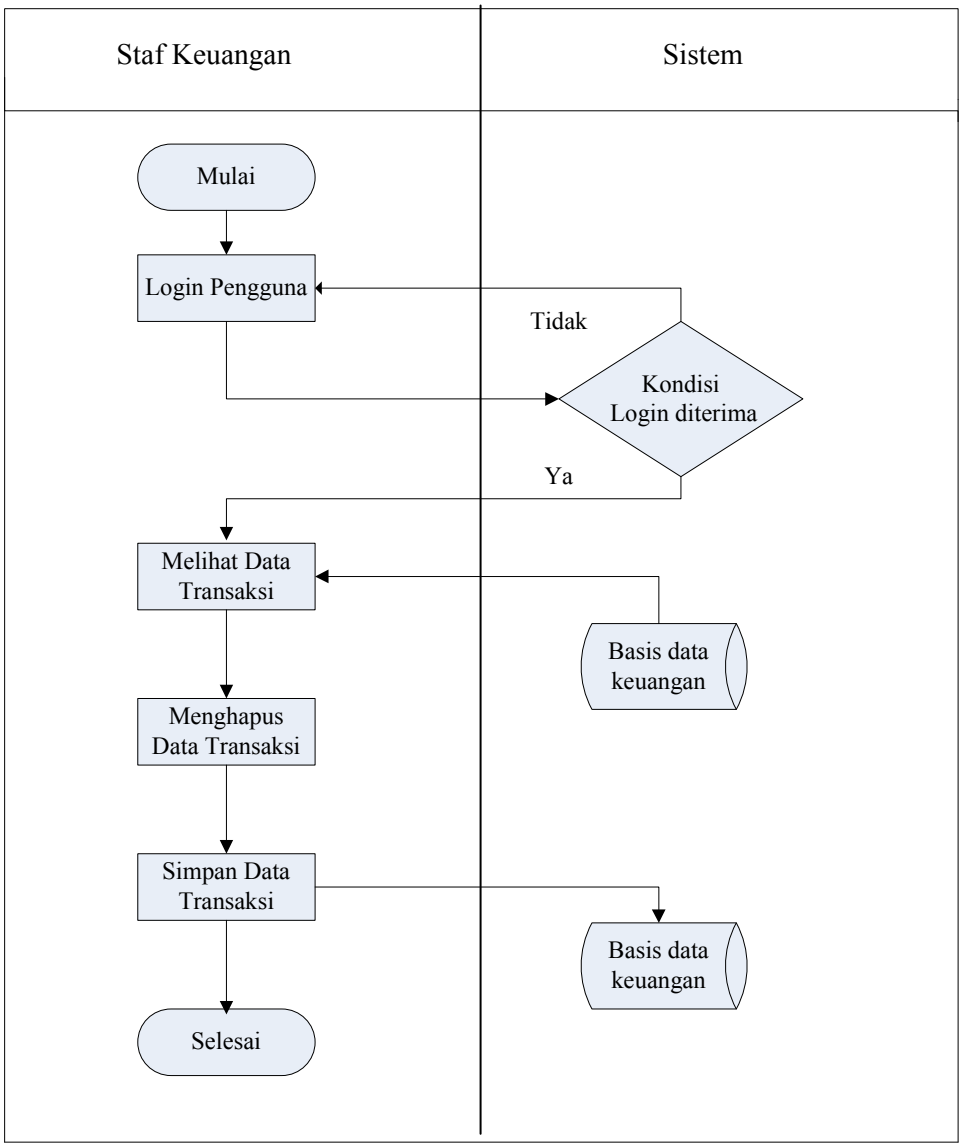
Gambaran aplikasi keuangan yang sedang berjalan dapat dilihat pada *flowchart* aplikasi keuangan pada Gambar 4.1, Gambar 4.2 dan Gambar 4.3.



Gambar 4.1 *Flowchart* Aplikasi Keuangan untuk Proses Simpan Transaksi



Gambar 4.2 *Flowchart* Aplikasi Keuangan untuk Proses Ubah Transaksi



Gambar 4.3 *Flowchart* Aplikasi Keuangan untuk Proses Hapus Transaksi

4.1.2 Basis Data pada Aplikasi Keuangan RSUD Bangkinang

Basis data yang dipergunakan pada aplikasi keuangan RSUD Bangkinang diberi nama dbKeuangan, dengan struktur tabel yang dapat dilihat pada Tabel 4.1.

Tabel 4.1 Daftar Tabel dalam dbKeuangan

No	Nama Tabel	Nama Kolom	Tipe Data	Ukuran	Nulls
1	2	3	4	5	6
1	tabel_akses	nama_akses	Char	20	Not Nulls
		kata_kunci	Char	30	Not Nulls
		nama_pengguna	Char	30	Allow Nulls
		hak_akses	Char	15	Not Nulls
		Status	Char	15	Allow Nulls
2	tabel_anggaran	kode_belanja	Char	5	Not Nulls
		no_program	Char	2	Not Nulls
		no_kegiatan	Char	5	Not Nulls
		kode_rekening	Char	15	Not Nulls
		jumlah_anggaran	numeric	18	Allow Nulls
		penerimaan_0	numeric	18	Allow Nulls
		pengeluaran_0	numeric	18	Allow Nulls
		penerimaan_2	numeric	18	Allow Nulls
		pengeluaran_2	numeric	18	Allow Nulls
		penerimaan_3	numeric	18	Allow Nulls
		pengeluaran_3	numeric	18	Allow Nulls
		penerimaan_0_lalu	numeric	18	Allow Nulls
		pengeluaran_0_lalu	numeric	18	Allow Nulls
		penerimaan_2_lalu	numeric	18	Allow Nulls
		pengeluaran_2_lalu	numeric	18	Allow Nulls
		penerimaan_3_lalu	numeric	18	Allow Nulls
		pengeluaran_3_lalu	numeric	18	Allow Nulls
3	tabel_belanja	kode_belanja	Char	5	Not Nulls
		nama_belanja	Char	20	Not Nulls
4	tabel_bku	no_bku	Integer	4	Not Nulls
		tanggal_transaksi	smalldatetime	4	Allow Nulls
		jenis_belanja	Char	2	Allow Nulls
		jenis_transaksi	Char	2	Allow Nulls
		kode_belanja	Char	5	Allow Nulls
		no_program	Char	2	Allow Nulls
		no_kegiatan	Char	5	Allow Nulls
		kode_rekening	Char	15	Allow Nulls
		Uraian	Char	1000	Allow Nulls
		jumlah_transaksi	Numeric	18	Allow Nulls
		flag_tutup_buku	Char	10	Allow Nulls
5	tabel_buku_bank	tanggal_transaksi	smalldatetime	4	Allow Nulls
		Uraian	Char	20	Allow Nulls

		penerimaan	Varchar	15	Allow Nulls
		Pengeluaran	Varchar	15	Allow Nulls
6	tabel_kegiatan	no_program	Char	2	Not Nulls
		no_kegiatan	Char	5	Not Nulls
		nama_kegiatan	Char	100	Allow Nulls
7	tabel_program	kode_belanja	Char	5	Not Nulls
		no_program	Char	2	Not Nulls
		nama_program	Char	80	Allow Nulls
8	tabel_rekening	kode_rekening	Char	15	Not Nulls
		nama_rekening	Char	80	Allow Nulls
9	tabel_setting	SKPD	Char	30	Not Nulls
		Kepala SKPD	Char	30	Allow Nulls
		Bendahara_Pengeluaran	Char	30	Allow Nulls
		Tahun_Anggaran	Char	8	Allow Nulls
		NIP	Char	20	Allow Nulls
		Pangkat	Char	20	Allow Nulls
		Jabatan	Char	30	Allow Nulls

4.2 Analisis Sistem Baru

Analisis sistem baru dilakukan untuk mendapatkan solusi dari kelemahan yang terdapat pada sistem lama, yaitu masalah keamanan basis data. Sistem yang akan dibangun akan menerapkan keamanan pada aplikasi keuangan RSUD Bangkinang, dengan menerapkan enkripsi menggunakan RC6 pada aplikasi keuangan. Aplikasi yang akan dikembangkan akan menambahkan fungsi-fungsi sebagai berikut :

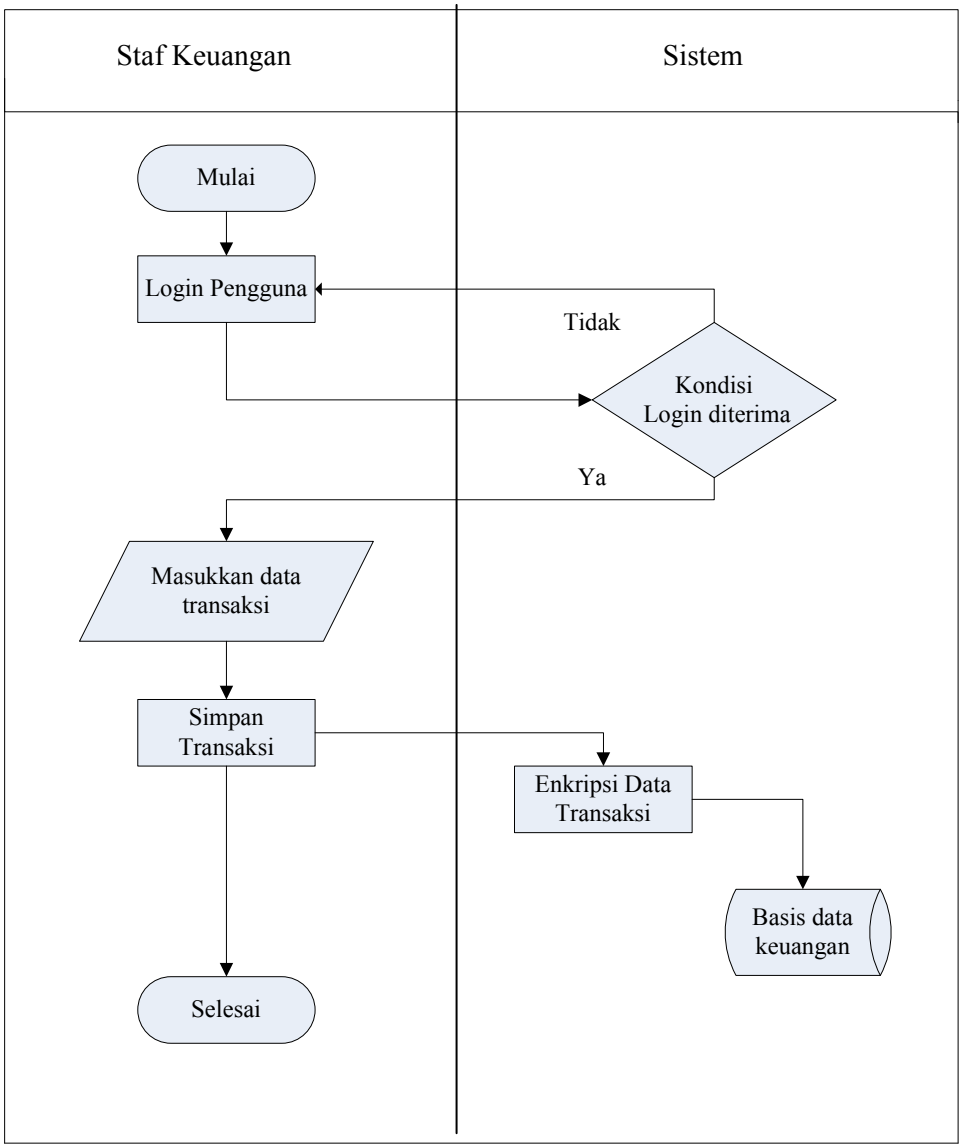
1. Melakukan enkripsi data yang akan disimpan ke dalam basis data sehingga data yang tersimpan dalam bentuk terenkripsi.
2. Melakukan deskripsi data yang tersimpan dalam basis data kedalam bentuk plainteks agar dapat dibaca dan diolah dalam aplikasi.

Seperti yang dijelaskan pada bab sebelumnya, dalam pengamanan basis data menggunakan enkripsi, terdapat dua strategi yang dapat digunakan, yaitu

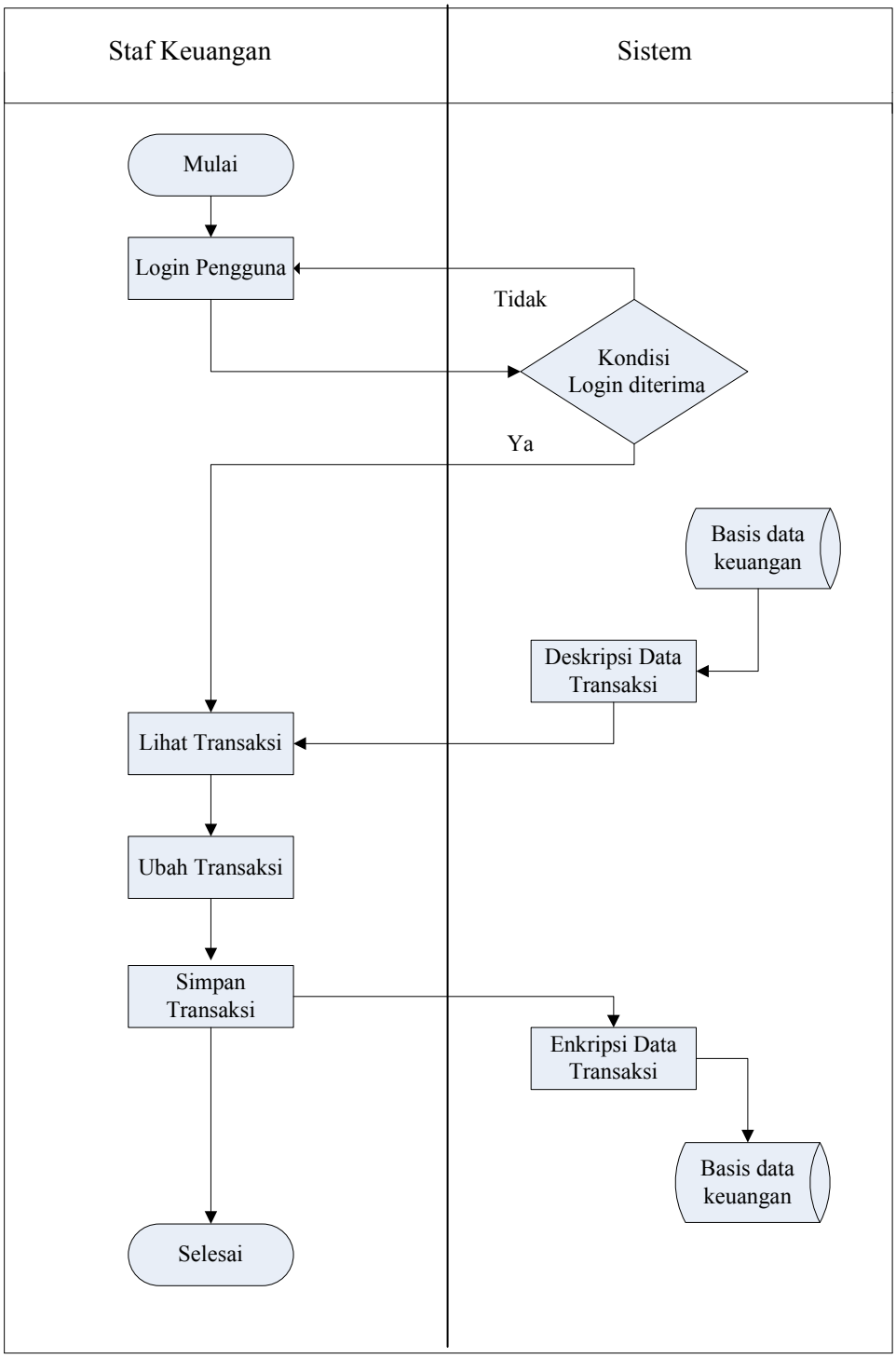
dengan enkripsi secara internal dalam basis data dan dengan melakukan enkripsi secara eksternal di luar basis data.

Dalam pelaksanaan tugas akhir ini, strategi yang digunakan adalah enkripsi secara eksternal diluar basis data, karena kelebihan dari strategi ini yaitu sistem pengamanan kunci yang baik dan peningkatan performa karena server basis data tidak dibebani dengan fungsi enkripsi

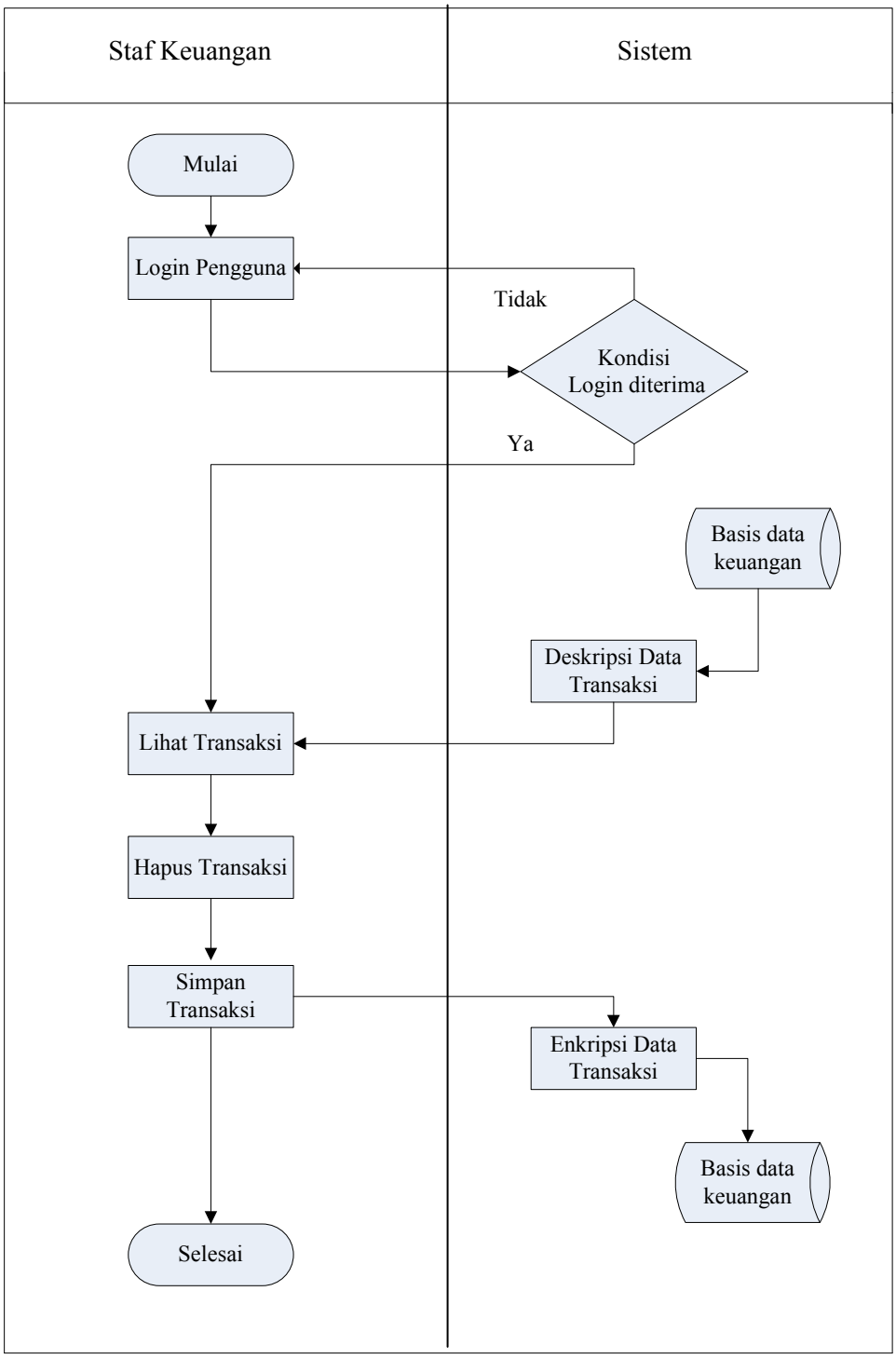
Gambaran sistem baru yang akan dikembangkan dengan mengimplementasikan algoritma RC6 dapat dilihat pada *flowchart* sistem baru pada Gambar 4.4, Gambar 4.5 dan Gambar 4.6.



Gambar 4.4 *Flowchart* Sistem Baru untuk Proses Simpan Transaksi



Gambar 4.5 *Flowchart* Sistem Baru untuk Proses Ubah Transaksi



Gambar 4.6 *Flowchart* Sistem Baru untuk Proses Hapus Transaksi

4.2.1 Analisis Kebutuhan

Pada subbab ini akan dilakukan analisis dari proses pembangunan perangkat lunak berdasarkan analisis yang telah dilakukan sebelumnya. Analisis yang akan dibahas meliputi meliputi analisis *input*, proses dan analisis *output*.

4.2.1.1 Analisis *Input*

Field yang menjadi *input* untuk proses enkripsi dan deskripsi adalah :

1. *Field-field* pada aplikasi keuangan yang akan disimpan ke basis data menjadi *input* pada proses enkripsi sebagai data yang akan dienkripsi.
2. *Field-field* pada basis data yang dimintakan oleh aplikasi untuk dimanipulasi menjadi *input* pada proses deskripsi.
2. Kunci pengguna menjadi *input* pada modul pembentukan kunci internal RC6.

4.2.1.2 Analisis Proses

Langkah-langkah yang dilakukan dalam proses penyelesaian masalah berupa :

1. Proses enkripsi data dilakukan ketika aplikasi akan menyimpan data transaksi, dalam hal ini data yang masih berbentuk teks asli (*plaintext*) kedalam basis data. Setelah enkripsi dilakukan, selanjutnya data dalam bentuk terenkripsi tersebut disimpan kedalam basis data.

2. Proses deskripsi dilakukan ketika aplikasi menjalankan perintah *select*, maka data yang terenkripsi dalam basis data dikembalikan ke bentuk plainteks, sehingga data dapat dilihat dan diolah oleh *user* pada level aplikasi.

4.2.1.3 Analisis *Output*

Hasil yang diharapkan dari sistem ini adalah data yang tersimpan pada basis data, pada tabel-tabel yang disebutkan diatas berupa data yang terenkripsi.

4.2.2 Analisis Basis Data pada Sistem Baru

Berikut akan dijelaskan analisis basis data pada sistem baru, meliputi analisis tabel yang perlu dienkripsi dan tidak perlu dienkripsi, serta perubahan struktur tabel_bku.

4.2.2.1 Analisis Tabel yang Perlu Dienkripsi dan Tidak Perlu Dienkripsi

Hal yang perlu diperhatikan dalam penerapan enkripsi pada basis data adalah menentukan data yang perlu dienkripsi atau tidak, karena tidak semua data yang disimpan dalam basis data perlu diproteksi dengan enkripsi, sebab bila semua data dienkripsi akan dapat mengurangi performansi basis data.

Tabel yang tidak dienkripsi adalah tabel_anggaran, tabel_program, tabel_kegiatan, tabel_rekening dan tabel_setting, karena isi dari tabel-tabel tersebut merupakan isi dari Dokumen Pelaksanaan Anggaran (DPA) RSUD Bangkinang, sehingga apabila *field* dari tabel-tabel tersebut ikut dienkripsi, akan membuka jalan bagi penyerang untuk menemukan kunci enkripsi dengan cara

membandingkan plainteks yang terdapat dalam DPA dan chiperteks yang terdapat dalam tabel.

Tabel yang akan dienkripsi meliputi tabel_bku dan tabel_buku_bank. Alasan pemilihan tabel-tabel tersebut karena tabel-tabel ini merupakan tempat penyimpanan data transaksi keuangan yang merupakan data-data penting dan sensitif pada aplikasi keuangan. Tabel_akses juga merupakan tabel yang dienkripsi karena tabel ini menyimpan data akses yang perlu diproteksi yaitu nama_akses, nama_pengguna, kata_kunci, hak_akses dan status.

Maka pada tabel-tabel tersebut, tipe datanya akan diubah agar dapat menampung chiperteks yang terdiri dari karakter-karakter khusus, sehingga tipe data pada tabel tersebut diubah menjadi *nvarchar*.

4.2.2.2 Perubahan Struktur Tabel_bku

Pada tabel_bku, terdapat kelemahan yaitu data pada baris tertentu mempunyai kesamaan dengan data pada baris yang lain (*redundan*). Sebagai contoh dapat dilihat pada Gambar 4.7, dimana pada kolom no_bku, tanggal_transaksi, jenis_belanja, jenis_transaksi, kode_belanja, no_program, no_kegiatan dan kode_rekening mempunyai kemungkinan data yang sama pada baris-barisnya.

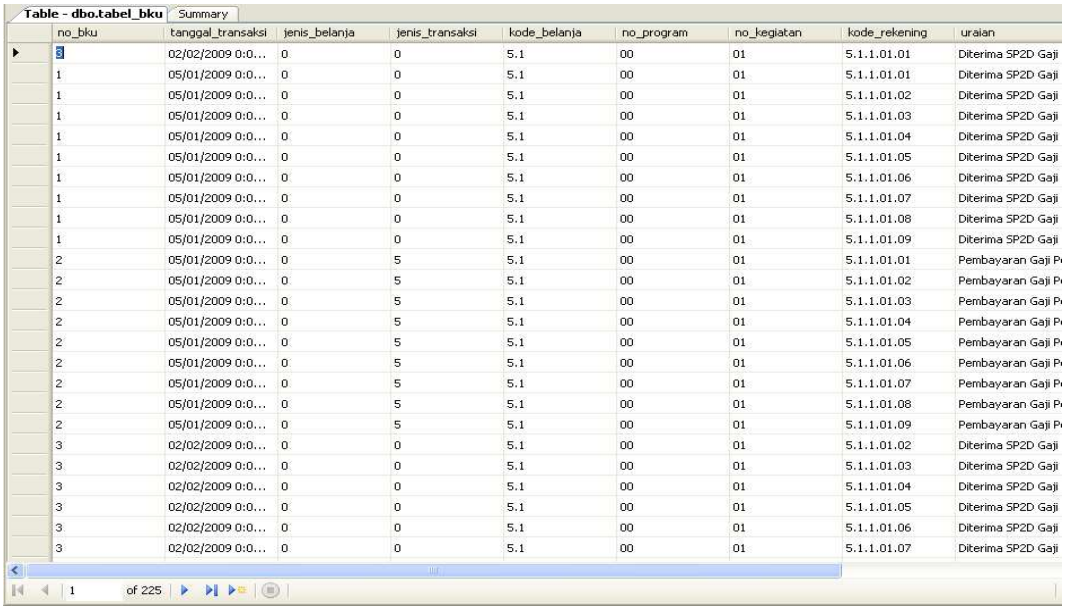


Table - dbo.tabel_bku | Summary

no_bku	tanggal_transaksi	jenis_belanja	jenis_transaksi	kode_belanja	no_program	no_kegiatan	kode_rekening	uraian
3	02/02/2009 0:0...	0	0	5.1	00	01	5.1.1.01.01	Diterima SP2D Gaji
1	05/01/2009 0:0...	0	0	5.1	00	01	5.1.1.01.01	Diterima SP2D Gaji
1	05/01/2009 0:0...	0	0	5.1	00	01	5.1.1.01.02	Diterima SP2D Gaji
1	05/01/2009 0:0...	0	0	5.1	00	01	5.1.1.01.03	Diterima SP2D Gaji
1	05/01/2009 0:0...	0	0	5.1	00	01	5.1.1.01.04	Diterima SP2D Gaji
1	05/01/2009 0:0...	0	0	5.1	00	01	5.1.1.01.05	Diterima SP2D Gaji
1	05/01/2009 0:0...	0	0	5.1	00	01	5.1.1.01.06	Diterima SP2D Gaji
1	05/01/2009 0:0...	0	0	5.1	00	01	5.1.1.01.07	Diterima SP2D Gaji
1	05/01/2009 0:0...	0	0	5.1	00	01	5.1.1.01.08	Diterima SP2D Gaji
1	05/01/2009 0:0...	0	0	5.1	00	01	5.1.1.01.09	Diterima SP2D Gaji
2	05/01/2009 0:0...	0	5	5.1	00	01	5.1.1.01.01	Pembayaran Gaji P
2	05/01/2009 0:0...	0	5	5.1	00	01	5.1.1.01.02	Pembayaran Gaji P
2	05/01/2009 0:0...	0	5	5.1	00	01	5.1.1.01.03	Pembayaran Gaji P
2	05/01/2009 0:0...	0	5	5.1	00	01	5.1.1.01.04	Pembayaran Gaji P
2	05/01/2009 0:0...	0	5	5.1	00	01	5.1.1.01.05	Pembayaran Gaji P
2	05/01/2009 0:0...	0	5	5.1	00	01	5.1.1.01.06	Pembayaran Gaji P
2	05/01/2009 0:0...	0	5	5.1	00	01	5.1.1.01.07	Pembayaran Gaji P
2	05/01/2009 0:0...	0	5	5.1	00	01	5.1.1.01.08	Pembayaran Gaji P
2	05/01/2009 0:0...	0	5	5.1	00	01	5.1.1.01.09	Pembayaran Gaji P
3	02/02/2009 0:0...	0	0	5.1	00	01	5.1.1.01.02	Diterima SP2D Gaji
3	02/02/2009 0:0...	0	0	5.1	00	01	5.1.1.01.03	Diterima SP2D Gaji
3	02/02/2009 0:0...	0	0	5.1	00	01	5.1.1.01.04	Diterima SP2D Gaji
3	02/02/2009 0:0...	0	0	5.1	00	01	5.1.1.01.05	Diterima SP2D Gaji
3	02/02/2009 0:0...	0	0	5.1	00	01	5.1.1.01.06	Diterima SP2D Gaji
3	02/02/2009 0:0...	0	0	5.1	00	01	5.1.1.01.07	Diterima SP2D Gaji

1 of 225

Gambar 4.7 Perulangan Data pada Beberapa Kolom pada Tabel_bku

Data ini nantinya akan menjadi blok plainteks pada proses enkripsi. Blok plainteks yang sama akan dienkripsi menjadi blok chiperteks yang sama (atau identik) (Munir, 2006). Hal ini dapat dimanfaatkan penyerang untuk menemukan plainteks maupun kunci enkripsi, dengan menggunakan serangan yang berbasis statistik (menggunakan frekuensi kemunculan blok chiper).

Solusi untuk masalah ini adalah dengan menggunakan ukuran blok yang besar dengan tujuan menghilangkan kemungkinan menghasilkan blok-blok yang identik. Solusi ini akan diterapkan pada tabel_bku, dengan cara mengubah struktur tabel, sehingga kolom-kolom yang mengandung nilai yang berulang tersebut digabungkan menjadi satu kolom, sehingga memperkecil kemungkinan plainteks yang sama. Maka kolom jenis_belanja, jenis_transaksi, kode_belanja, no_program, no_kegiatan dan kode_rekening akan digabungkan menjadi kolom yang diberi nama uraian_belanja.

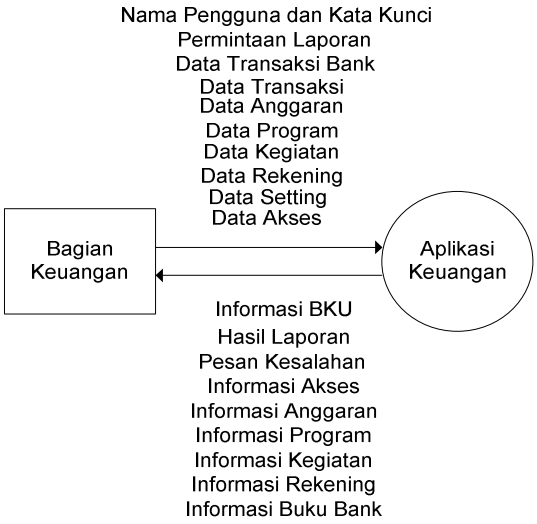
4.2.3 Analisis Fungsional

Analisis fungsional adalah penggambaran sistem secara umum. Berdasarkan batasan masalah dalam pelaksanaan tugas akhir ini, yaitu menerapkan algoritma RC6 pada aplikasi keuangan, maka terdapat perubahan pada *Context Diagram* dan DFD dari sistem lama, dengan menambahkan proses enkrip dan dekrip. *Context Diagram* dan DFD sistem lama secara lengkap dapat dilihat pada lampiran A.

Berikut ini adalah *Context Diagram* dan DFD baru yang menggambarkan perubahan dan penambahan proses enkrip dan dekrip.

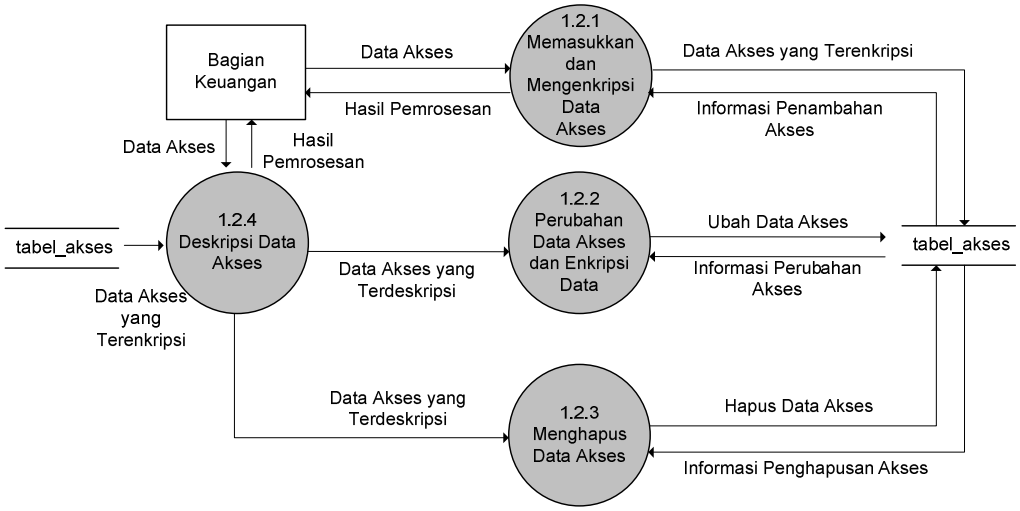
4.2.3.1 *Context Diagram* Aplikasi Keuangan

Context Diagram adalah diagram yang merepresentasikan seluruh elemen sistem sebagai sebuah *bubble* tunggal dengan data *input* dan *output* yang ditunjukkan oleh anak panah yang masuk dan keluar secara berurutan. *Context Diagram* Aplikasi Keuangan RSUD Bangkinang dapat dilihat pada Gambar 4.8.



Gambar 4.8 *Context Diagram* Aplikasi Keuangan RSUD Bangkinang

4.2.3.2 Perubahan pada DFD Level 3 Pengelolaan Data Akses



Gambar 4.9 DFD Level 3 Pengelolaan Data Akses

Tabel 4.2 Keterangan Proses pada DFD Sistem Baru Level 3 Proses 1.2.1

No. Proses	:	1.2.1
Nama Proses	:	Memasukkan dan mengenkripsi data akses
Deskripsi	:	Memasukkan dan mengenkripsi data akses
Input	:	Data akses
Output	:	Penambahan data akses yang terenkripsi
Logika Proses	:	Menambah dan mengenkripsi data akses

Tabel 4.3. Keterangan Proses pada DFD Sistem Baru Level 3 Proses 1.2.2

No. Proses	:	1.2.2
Nama Proses	:	Mengubah dan mengenkripsi data akses
Deskripsi	:	Mengubah dan mengenkripsi data akses
Input	:	Data akses yang terdeskripsi
Output	:	Pengubahan data akses dan mengenkripsi kembali
Logika Proses	:	Mengubah dan mengenkripsi data akses

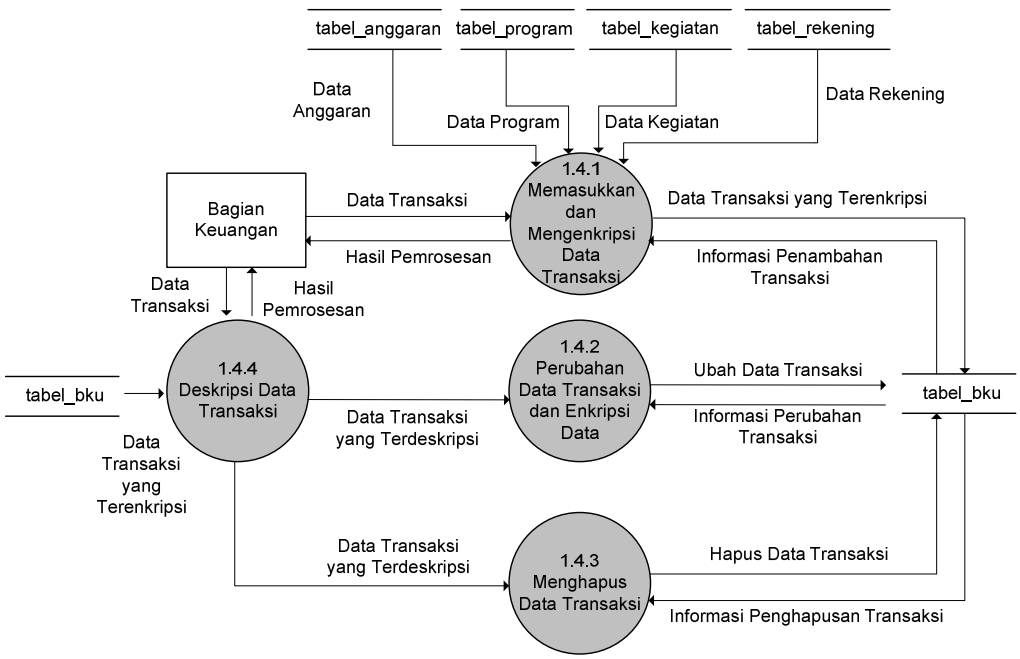
Tabel 4.4 Keterangan Proses pada DFD Sistem Baru Level 3 Proses 1.2.3

No. Proses	:	1.2.3
Nama Proses	:	Menghapus data akses
Deskripsi	:	Menghapus data akses
Input	:	Data akses yang terdeskripsi
Output	:	Penghapusan data akses
Logika Proses	:	Menghapus data akses

Tabel 4.5 Keterangan Proses pada DFD Sistem Baru Level 3 Proses 1.2.4

No. Proses	:	1.2.4
Nama Proses	:	Deskripsi data akses
Deskripsi	:	Mendeskripsi data akses
Input	:	Data akses yang terenkripsi
Output	:	Pendeskripsian data akses yang terenkripsi
Logika Proses	:	Melakukan deskripsi data akses

4.2.3.3 Perubahan pada DFD Level 3 Pengelolaan Data Transaksi



Gambar 4.10 DFD Level 3 Pengelolaan Data Transaksi

Tabel 4.6 Keterangan Proses pada DFD Sistem Baru Level 3 Proses 1.4.1

No. Proses	:	1.4.1
Nama Proses	:	Memasukkan dan mengenkripsi data transaksi
Deskripsi	:	Memasukkan dan mengenkripsi data transaksi
Input	:	Data transaksi, data anggaran, program, kegiatan dan rekening
Output	:	Penambahan data transaksi yang terenkripsi
Logika Proses	:	Menambah dan mengenkripsi data transaksi

Tabel 4.7 Keterangan Proses pada DFD Sistem Baru Level 3 Proses 1.4.2

No. Proses	:	1.4.2
Nama Proses	:	Mengubah dan mengenkripsi data transaksi
Deskripsi	:	Mengubah dan mengenkripsi data transaksi
Input	:	Data transaksi yang terdeskripsi
Output	:	Pengubahan data transaksi dan mengenkripsi kembali
Logika Proses	:	Mengubah dan mengenkripsi data transaksi

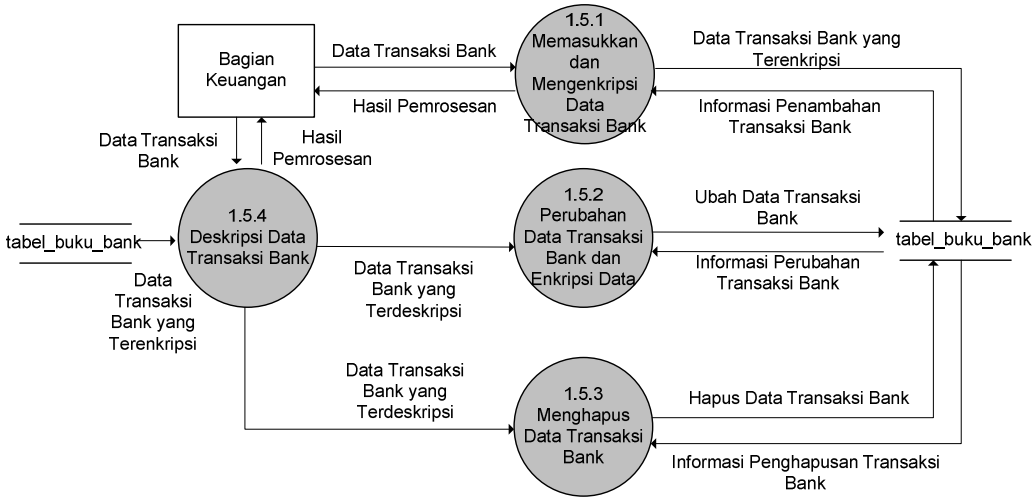
Tabel 4.8 Keterangan Proses pada DFD Sistem Baru Level 3 Proses 1.4.3

No. Proses	:	1.4.3
Nama Proses	:	Menghapus data transaksi
Deskripsi	:	Menghapus data transaksi
Input	:	Data transaksi yang terdeskripsi
Output	:	Penghapusan data transaksi
Logika Proses	:	Menghapus data transaksi

Tabel 4.9 Keterangan Proses pada DFD Sistem Baru Level 3 Proses 1.4.4

No. Proses	:	1.4.4
Nama Proses	:	Deskripsi data transaksi
Deskripsi	:	Mendeskripsi data transaksi
Input	:	Data transaksi yang terenkripsi
Output	:	Pendeskripsian data transaksi yang terenkripsi
Logika Proses	:	Melakukan deskripsi data transaksi

4.2.3.4 Perubahan pada DFD Level 3 Pengelolaan Data Buku Bank



Gambar 4.11 DFD Level 3 Pengelolaan Data Buku Bank

Tabel 4.10 Keterangan Proses pada DFD Sistem Baru Level 3 Proses 1.5.1

No. Proses	:	1.5.1
Nama Proses	:	Memasukkan dan mengenkripsi data transaksi bank
Deskripsi	:	Memasukkan dan mengenkripsi data transaksi bank
Input	:	Data transaksi bank
Output	:	Penambahan data transaksi bank yang terenkripsi
Logika Proses	:	Menambah dan mengenkripsi data transaksi bank

Tabel 4.11 Keterangan Proses pada DFD Sistem Baru Level 3 Proses 1.5.2

No. Proses	:	1.5.2
Nama Proses	:	Mengubah dan mengenkripsi data transaksi bank
Deskripsi	:	Mengubah dan mengenkripsi data transaksi bank
Input	:	Data transaksi bank yang terdeskripsi
Output	:	Pengubahan data transaksi bank dan mengenkripsi kembali
Logika Proses	:	Mengubah dan mengenkripsi data transaksi bank

Tabel 4.12 Keterangan Proses pada DFD Sistem Baru Level 3 Proses 1.5.3

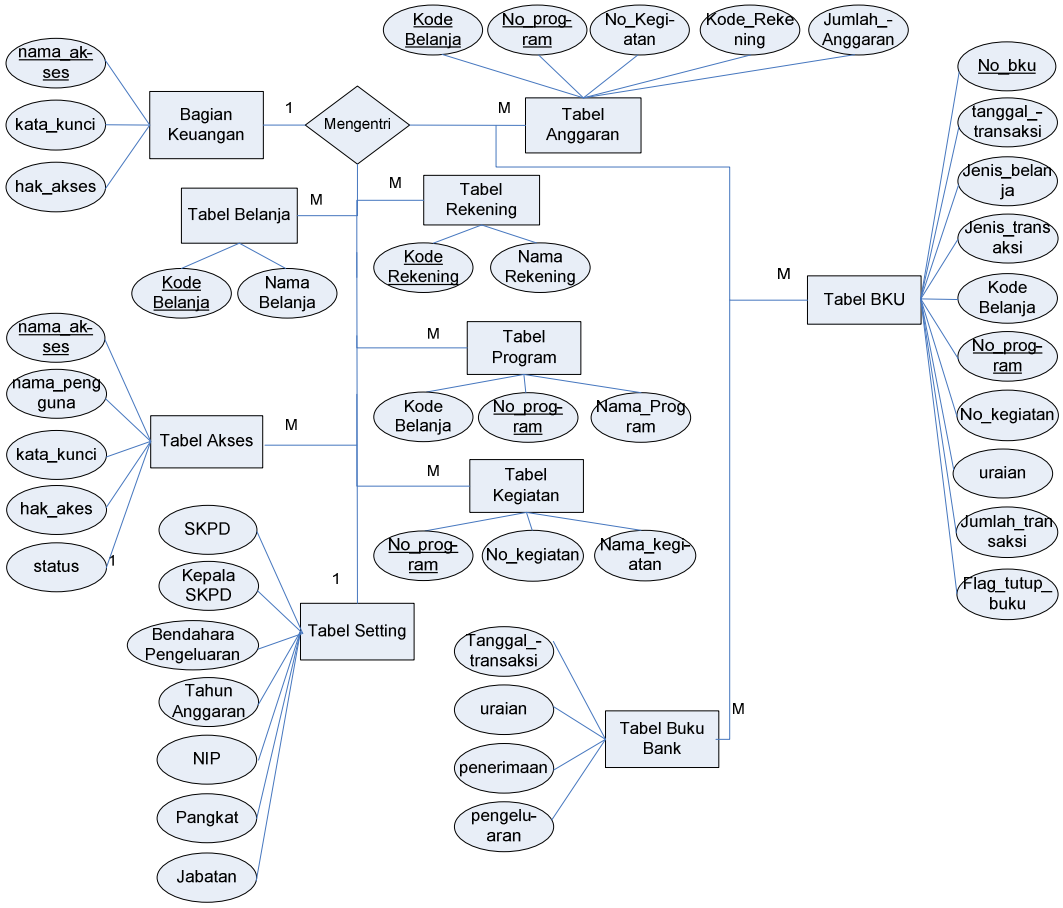
No. Proses	:	1.5.3
Nama Proses	:	Menghapus data transaksi
Deskripsi	:	Menghapus data transaksi
Input	:	Data transaksi yang terdeskripsi
Output	:	Penghapusan data transaksi
Logika Proses	:	Menghapus data transaksi

Tabel 4.13 Keterangan Proses pada DFD Sistem Baru Level 3 Proses 1.5.4

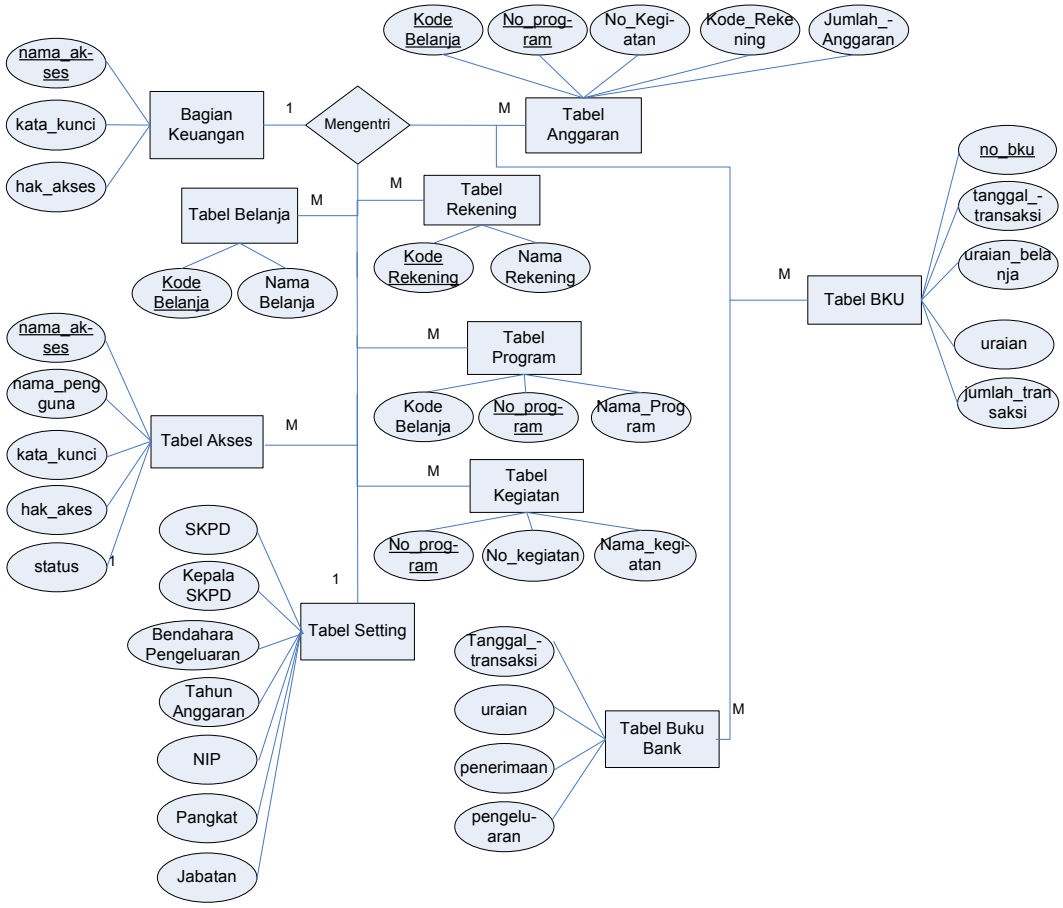
No. Proses	:	1.5.4
Nama Proses	:	Deskripsi data transaksi bank
Deskripsi	:	Mendeskripsi data transaksi bank
Input	:	Data transaksi bank yang terenkripsi
Output	:	Pendeskripsian data transaksi bank yang terenkripsi
Logika Proses	:	Melakukan deskripsi data transaksi bank

4.2.4 Analisis Data

Dalam melakukan analisis data akan digunakan ER Diagram. *Entity Relationship Diagram* berfungsi untuk menggambarkan diagram keterhubungan antar objek data didalam sistem. ERD sistem lama dapat dilihat pada Gambar 4.12, sedangkan ERD sistem baru dapat dilihat pada Gambar 4.13. Perbedaan ERD sistem lama dan sistem baru terdapat pada field tabel_bku, dimana field tabel_bku pada sistem baru mengikuti analisis basis data pada sistem baru.



Gambar 4.12 ERD Sistem Lama



Gambar 4.13 ERD Sistem Baru

Kamus Data :

Tabel 4.14 Kamus Data Bagian Keuangan

Nama	:	Bagian Keuangan
Deskripsi	:	nama_akses, kata_kunci, hak_akses
Bentuk Data	:	Tabel atau file
Sumber/Tujuan	:	Berasal dari *user* yang berwenang sebagai bagian keuangan agar dapat masuk dan menggunakan aplikasi dengan aman.
Periode	:	Setiap dilakukannya proses *login*
Volume	:	Tergantung *input* dari bagian keuangan
Struktur Data	:	# nama_akses + kata_kunci

Tabel 4.15 Kamus Data Tabel Akses

Nama	:	Tabel Akses
Deskripsi	:	Hasil *input* dari data akses
Bentuk Data	:	Tabel atau file
Sumber/Tujuan	:	Berasal dari bagian keuangan agar bagian keuangan mempunyai arsip data akses.
Periode	:	Setiap dilakukannya proses *input* data akses
Volume	:	Tergantung *input* dari bagian keuangan
Struktur Data	:	# nama_akses + nama_pengguna + kata_kunci + hak_akses + status

Tabel 4.16 Kamus Data Tabel Anggaran

Nama	:	Tabel Anggaran
Deskripsi	:	Hasil *input* dari data anggaran
Bentuk Data	:	Tabel atau file
Sumber/Tujuan	:	Berasal dari perbendaraan agar bagian keuangan dan bagian keuangan mempunyai arsip data anggaran.
Periode	:	Setiap dilakukannya proses *input* data anggaran
Volume	:	Tergantung *input* dari bagian keuangan
Struktur Data	:	# kode_belanja + no_program + no_kegiatan + kode_rekening + jumlah_anggaran

Tabel 4.17 Kamus Data Tabel Belanja

Nama	:	Tabel Belanja
Deskripsi	:	Hasil *input* dari data belanja
Bentuk Data	:	Tabel atau file
Sumber/Tujuan	:	Berasal dari perbendaraan agar bagian keuangan mempunyai arsip data belanja.
Periode	:	Setiap dilakukannya proses *input* data belanja
Volume	:	Tergantung *input* dari bagian keuangan
Struktur Data	:	# kode_belanja + nama_belanja

Tabel 4.18 Kamus Data Tabel Program

Nama	:	Tabel Program
Deskripsi	:	Hasil *input* dari data program
Bentuk Data	:	Tabel atau file
Sumber/Tujuan	:	Berasal dari perbendaraan agar bagian keuangan mempunyai arsip data program.
Periode	:	Setiap dilakukannya proses *input* data program
Volume	:	Tergantung *input* dari bagian keuangan
Struktur Data	:	# kode_belanja + no_program + nama_program

Tabel 4.19 Kamus Data Tabel Kegiatan

Nama	:	Tabel Kegiatan
Deskripsi	:	Hasil *input* dari data kegiatan
Bentuk Data	:	Tabel atau file
Sumber/Tujuan	:	Berasal dari perbendaraan agar bagian keuangan mempunyai arsip data kegiatan.
Periode	:	Setiap dilakukannya proses *input* data kegiatan
Volume	:	Tergantung *input* dari bagian keuangan
Struktur Data	:	# no_program + no_kegiatan + nama_kegiatan

Tabel 4.20 Kamus Data Tabel Rekening

Nama	:	Tabel Rekening
Deskripsi	:	Hasil *input* dari data rekening
Bentuk Data	:	Tabel atau file
Sumber/Tujuan	:	Berasal dari perbendaraan agar bagian keuangan mempunyai arsip data rekening.
Periode	:	Setiap dilakukannya proses *input* data rekening
Volume	:	Tergantung *input* dari bagian keuangan
Struktur Data	:	# kode_rekening + nama_rekening

Tabel 4.21 Kamus Data Tabel Setting

Nama	:	Tabel Setting
Deskripsi	:	Hasil *input* dari data setting
Bentuk Data	:	Tabel atau file
Sumber/Tujuan	:	Berasal dari perbendaraan agar bagian keuangan mempunyai arsip data setting.
Periode	:	Setiap dilakukannya proses *input* data setting
Volume	:	Tergantung *input* dari bagian keuangan
Struktur Data	:	#SKPD + Kepala_SKPD + Bendahara_Pengeluaran + Tahun_Anggaran + NIP + Pangkat + Jabatan

Tabel 4.22 Kamus Data Tabel BKU Sistem Lama

Nama	:	Tabel BKU
Deskripsi	:	Hasil *input* dari data transaksi keuangan
Bentuk Data	:	Tabel atau file
Sumber/Tujuan	:	Berasal dari bagian keuangan agar bagian keuangan mempunyai arsip data transaksi.
Periode	:	Setiap dilakukannya proses *input* data transaksi
Volume	:	Tergantung *input* dari bagian keuangan
Struktur Data	:	# no_bku + tanggal_transaksi + jenis_belanja + jenis_transaksi + kode_belanja + no_program + no_kegiatan + uraian_belanja + uraian + jumlah transaksi

Tabel 4.23 Kamus Data Tabel Buku Bank

Nama	:	Tabel Buku Bank
Deskripsi	:	Hasil *input* dari data transaksi bank
Bentuk Data	:	Tabel atau file
Sumber/Tujuan	:	Berasal dari bagian keuangan agar bagian keuangan mempunyai arsip data transaksi bank
Periode	:	Setiap dilakukannya proses *input* data transaksi bank
Volume	:	Tergantung *input* dari bagian keuangan
Struktur Data	:	# tanggal_transaksi + uraian + penerimaan + pengeluaran

Tabel 4.24 Kamus Data Tabel BKU Sistem Baru

Nama	:	Tabel BKU
Deskripsi	:	Hasil *input* dari data transaksi keuangan
Bentuk Data	:	Tabel atau file
Sumber/Tujuan	:	Berasal dari bagian keuangan agar bagian keuangan mempunyai arsip data transaksi.
Periode	:	Setiap dilakukannya proses *input* data transaksi
Volume	:	Tergantung *input* dari bagian keuangan
Struktur Data	:	# no_bku + tanggal_transaksi + uraian_belanja + uraian + jumlah transaksi

4.3 Analisis Penerapan Algoritma RC6 dalam Enkripsi Basis Data

Secara lengkap, algoritma RC6 ditulis sebagai berikut:

RC6 –w/r/b

Algoritma RC6 memiliki beberapa parameter yaitu panjang *word* (*w*), jumlah iterasi (*r*) dan panjang kunci (*b*). Dengan adanya parameter tersebut, dalam melakukan implementasi algoritma RC6 dapat disesuaikan kebutuhan yang diinginkan. Apabila yang diinginkan adalah algoritma enkripsi yang cepat maka jumlah iterasi yang dilakukan dapat dikurangi, walaupun dengan mengurangi jumlah iterasi dapat mengurangi kekuatan keamanan algoritma RC6. Sebaliknya, jika menambahkan jumlah iterasi, maka akan dihasilkan algoritma kriptografi yang kuat, namun membutuhkan waktu yang cukup lama dalam melakukan enkripsi. Dan untuk menyesuaikan implementasi dengan mesin yang menjadi target implementasi dapat dilakukan dengan melakukan perubahan pada panjang *word*. Dengan karakteristik seperti ini, algoritma RC6 akan dapat diimplementasikan secara fleksibel pada berbagai jenis platform dan mesin dengan prosesor yang beragam.

Pada perangkat lunak yang akan dibangun, panjang blok yang akan digunakan sebesar 128 bit, artinya panjang *word* sebesar 32 bit. Panjang *word* tersebut dipilih karena pada dasarnya algoritma RC6 memang ditujukan untuk menggunakan panjang *word* tersebut dan mudah untuk diimplementasikan karena algoritma RC6 menggunakan operasi *integer modulo* sebesar panjang *word* dan tipe integer sebagian besar *compiler* yang beredar sekarang ini memiliki panjang 32 bit.

Karakter-karakter yang akan dienkripsi akan diubah kedalam nilai ASCII, dimana nilai karakter dalam tabel karakter ASCII ditambah tabel karakter spesial adalah 0 sampai dengan 255, artinya satu karakter dalam ASCII akan diwakili oleh 8 bit, dimana $2^8 = 256$. Sehingga, dalam 1 blok plainteks (32 bit) akan menyimpan 4 karakter dan setiap kali iterasi, maka akan diambil 16 karakter dari plainteks.

Apabila panjang plainteks atau panjang kunci kurang dari 16 karakter, maka akan dilakukan *padding*, yaitu dengan menambahkan karakter *space* (spasi) di akhir teks sehingga panjang teks mencukupi 16 karakter. Karakter spasi dipilih untuk mempermudah proses deskripsi, karena pada bahasa pemrograman yang akan dipilih terdapat fungsi *rtrim*, yang memiliki fungsi menghilangkan karakter spasi di akhir teks.

Algoritma RC6 yang akan digunakan dalam perangkat lunak yang akan dibangun, dengan w sebesar 32 bit, r sebesar 20 kali putaran dan panjang kunci sebesar 16 *byte*. Langkah-langkah algoritma RC6 dalam pelaksanaan tugas akhir ini akan dikelompokkan kedalam beberapa bagian, yaitu :

1. Pembangkit Sub Kunci

   Kunci dari pengguna menjadi masukan pada tahap pembangkit sub kunci. Kunci pengguna ini akan dimasukkan oleh pengguna pada saat akan melakukan proses enkripsi dan deskripsi. Kunci ini memiliki tipe data *string* dan memiliki panjang 16 *byte* (16 karakter).

2. Baca Masukan untuk Proses Enkripsi

   Yang dilakukan pada tahap ini adalah membaca teks yang menjadi masukan pada proses enkripsi, yaitu *field* dari aplikasi keuangan.

   Pada proses simpan_transaksi, *field*nya adalah no_bku, tanggal_transaksi, uraian_belanja, uraian dan jumlah_transaksi.

   Pada proses simpan_transaksi_bank, *field*nya adalah tanggal_transaksi, uraian, penerimaan dan pengeluaran.

   Pada proses simpan_data_akses, *field*nya adalah nama_akses, nama_pengguna, kata_kunci, hak_akses dan status.

3. Enkripsi, meliputi *whitening* awal, iterasi dan *whitening* akhir.

4. Baca Masukan untuk Proses Deskripsi

   Yang dilakukan pada tahap ini adalah membaca teks yang menjadi masukan pada proses deskripsi, yaitu *record* dari tabel dalam basis data dbKeuangan.

   Pada proses lihat_bku, masukan adalah isi dari tabel_bku, yaitu kolom no_bku, tanggal_transaksi, uraian_belanja, uraian dan jumlah_transaksi.

   Pada proses lihat_transaksi_bank, masukan adalah isi dari tabel_buku_bank, yaitu kolom tanggal_transaksi, uraian, penerimaan dan pengeluaran.

Pada proses lihat_akses, masukan adalah isi dari tabel_akses, yaitu kolom nama_akses, nama_pengguna, kata_kunci, hak_akses dan status.

5. Deskripsi, merupakan kebalikan dari proses enkripsi.

Langkah-langkah diatas akan dijelaskan dalam algoritma-algoritma berikut :

4.3.1 Algoritma Pembangkit Sub Kunci

```
Kamus
      Type Word32 : 32 bit {tipe data 32 bit}
      Kunci : string {kunci yang dimasukkan oleh pengguna}
      i, j, c, s, v : integer
      A : integer
      B : integer
      S : array [0..43] of word32
      L : array [0..43] of word32
Function ROTL (X:Word32; Y : integer) → Word32 (fungsi untuk
      merotasi bit sebanyak variabel kedua)
Algoritma
      Input (Kunci)
      S[0] ← b7e15163
      For i ← 1 to 43 do
            S[i] ← S[i-1] + 9e3779b9
      Endfor
      A ← B ← i ← j ← 0
      v ← 44
      If (c > v) then
      v ← c
      v ← v*3
```

```
For s ← 1 to v do
      A ← S[i] ← ROTL ((S[i] + A + B), 3)
      B ← L[j] ← ROTL (L[j] + A + B, A + B)
      i ← (i+1) mod 44
      j ← (j+1) mod c
Endfor
```

4.3.2 Algoritma Baca Masukan untuk Proses Enkripsi

Prosedur Baca_Masukan_Proses_Enkripsi

{*Input* : Field masukan belum dibaca}

{*Output* : Field masukan dibaca per 16 karakter dan ditampung dalam buffer. Pada proses simpan_transaksi, fieldnya adalah no_bku, tanggal_transaksi, uraian_belanja, uraian dan jumlah_transaksi. Pada proses proses simpan_transaksi_bank, *field*nya adalah tanggal_transaksi, uraian, penerimaan dan pengeluaran. Pada proses simpan_data_akses, *field*nya adalah nama_akses, nama_pengguna, kata_kunci, hak_akses dan status.}

Kamus

```
field_masukan : string {field_masukan}
Buff : array [0..15] of char
i : integer
```

Algoritma

```
Input (field_masukan)
i ← 0
```

```
while (i <= 15) and not (EOF) do
      Read (field_masukan, Buff[i])
Endwhile
```

4.3.3 Algoritma *Whitening* Awal

```
Prosedur Whitening_awal {Input    : blok kedua dan keempat belum
                                    dijumlahkan dengan sub kunci
                         Output   : blok kedua dan keempat yang
                                    telah dijumlahkan dengan sub
                                    kunci}
Kamus
      Type word32 : 32 bit (tipe data sebesar 32 bit)
      X : word32 array [0..3] {blok enkripsi/plainteks}
      S : array [0..43] of word32 {sub kunci}
Algoritma
      X[1] ← X[1] + S[0]
      X[3] ← X[3] + S[1]
```

4.3.4 Algoritma Iterasi

```
Prosedur Iterasi  {Input    : keempat blok setelah whitening
                               awal belum diproses
                   Output   : keempat blok yang telah diproses
                               dan saling dipertukarkan}
Kamus
      Type word32 : 32 bit {tipe data sebesar 32 bit}
      X : word32 array [0..3] {blok enkripsi/plainteks}
      Function ROTL(X : Word32; Y : integer) → word32
      {Merotasi bit kekiri sebanyak variabel kedua}
```

```
      Temp : word32
      u, t: word32
      i : integer
Algoritma
      For i ← 1 to 20 do
            t ← ROTL ((X[1]*(2*X[1]+1)), 5)
            u ← ROTL ((X[3]*(2*X[3]+1)), 5)
            X[0] ← (ROTL((X[0] XOR t), u )) + S[2*i]
            X[2] ← (ROTL((X[2] XOR u), t )) + S[2*i + 1]
            Temp ← X[0]
            X[0] ← X[1]
            X[1] ← X[2]
            X[2] ← X[3]
            X[3] ← Temp
      End for
```

4.3.5 Algoritma *Whitening* Akhir

```
Prosedur Whitening_akhir {Input     : blok pertama dan ketiga
                                      belum dijumlahkan dengan sub
                                      kunci
                          Output    : blok pertama dan ketiga yang
                                      telah dijumlahkan dengan sub
                                      kunci}
Kamus
      Type word32 : 32 bit (tipe data sebesar 32 bit)
      X : word32 array [0..3] {blok enkripsi/plainteks}
      S : array [0..43] of word32 {sub kunci}
```

Algoritma

X[0] ← X[0] + S[42]

X[2] ← X[2] + S[43]

4.3.6 Algoritma Baca File Masukan untuk Proses Deskripsi

Prosedur Baca_File_Masukan_Proses_Deskripsi

{*Input* : Tabel pada basis data belum dibuka.
Field masukan berupa chiperteks belum dibaca. }

{*Output* : Field pada tabel dalam basis data dbKeuangan, yaitu tabel_bku, tabel_buku_bank dan tabel_akses, yang berupa chiperteks dibaca per 16 karakter dan ditampung dalam buffer.
Pada proses lihat_bku, field masukan adalah isi dari kolom tabel_bku, yaitu no_bku, tanggal_transaksi, uraian_belanja, uraian dan jumlah_transaksi.
Pada proses lihat_transaksi_bank, field masukan adalah isi kolom dari tabel_buku_bank, yaitu tanggal_transaksi, uraian, penerimaan dan pengeluaran.
Pada proses lihat_akses, field masukan adalah isi kolom dari tabel_akses, yaitu nama_akses, nama_pengguna, kata_kunci, hak_akses dan status. }

Kamus

Isi_kolom : string {isi_kolom}

Buff : array [0..15] of char

i : integer

Algoritma

```
Open (dbKeuangan)
Open (tabel)
While not tabel.EOF do
    Input (isi_kolom)
    i ← 0
    while (i <= 15) and not (isi_kolom.EOF) do
        Read (isi_kolom, Buff[i])
    Endwhile
Endwhile
```

4.3.7 Algoritma Deskripsi

```
Prosedur Deskripsi {Input  : keempat blok belum diproses
                    Output : keempat blok yang telah diproses
                             dan saling dipertukarkan}
Kamus
    Type word32 : 32 bit {tipe data sebesar 32 bit}
    X : word32 array [0..3] {blok enkripsi/plainteks}
    Function ROTL(X:Word32; Y : integer) → word32
    {Merotasi bit kekiri sebanyak variabel kedua}
    Temp : word32
    u, t: word32
    i : integer
Algoritma
    X[2] ← X[2] - S[43]
    X[0] ← X[0] - S[42]
    For i ← 20 downto 1 do
        Temp ← X[3]
        X[3] ← X[2]
```

```
        X[2] ← X[1]
        X[1] ← X[0]
        X[0] ← Temp
        u ← ROTL ((X[3]*(2*X[3]+1)), 5)
        t ← ROTL ((X[1]*(2*X[1]+1)), 5)
        X[2] ← (ROTR(X[2] - S[2*i + 1]), t ) XOR u)
        X[0] ← (ROTR(X[0] - S[2*i]), u ) XOR t)
    End for
    X[3] ← X[3] - S[1]
    X[1] ← X[1] - S[0]
```

4.3.8 Perhitungan Manual Algoritma RC6

Pada perhitungan manual algoritma RC6 ini diberikan kunci sebesar 16 byte dan plainteks sebesar 128 bit (16 *byte*). Kunci dan plainteks yang menjadi contoh masing-masing sebagai berikut :

Kunci : `syahrial12345678`

Plainteks : `teknik informati`

Langkah pertama adalah membagi plainteks kedalam 4 blok yaitu blok A, B, C dan D, yang masing-masing blok yang terdiri dari 32 bit (4 karakter).

```
 A     B     C     D
tekn  ik i  nfor  mati
```

Ubah tiap karakter dalam masing-masing blok kedalam nilai ASCII, selanjutnya ubah nilai ASCII tersebut menjadi bilangan biner masing- masing sepanjang 8 bit, sehingga pada masing-masing blok akan dihasilkan bilangan biner sepanjang 32 bit.

Blok A :

Plainteks	t	e	k	n
ASCII	116	101	107	110
Biner	01110100	01100101	01101011	01101110

Blok B :

Plainteks	i	k		i
ASCII	105	107	160	105
Biner	01101001	01101011	00100000	01101001

Blok C :

Plainteks	n	f	o	r
ASCII	110	102	111	114
Biner	01101110	01100110	01101111	01110010

Blok D :

Plainteks	m	a	t	i
ASCII	109	97	116	105
Biner	01101101	01100001	01110100	01101001

Kemudian bilangan biner digabungkan kembali, dengan aturan *byte* pertama plainteks diletakkan pada *least significant bit* blok A, dan *byte* terakhir plainteks diletakkan pada *most significant bit* blok D.

```
Blok A      : 01101110011010110110010101110100
              dalam desimal : 1.852.532.084.
Blok B      : 01101001001000000110101101101001
              dalam desimal : 1.763.732.329.
Blok C      : 01110010011011110110011001101110
              dalam desimal : 1.919.903.342.
Blok D      : 01101001011101000110000101101101
              dalam desimal : 1.769.234.797.
```

Setelah didapatkan nilai pada masing-masing blok, maka dilanjutkan dengan langkah-langkah berikut (Perhitungan manual pembangkit sub kunci dapat dilihat pada Lampiran B) :

1. Whitening Awal

Whitening awal, dengan menjumlahkan B dengan sub kunci S(0), dan D dengan sub kunci S(1). Penjumlahan dilakukan dalam modulo 2^{32}.

$B = B + S(0) \bmod 2^{32}$

$D = D + S(1) \bmod 2^{32}$

$B = 1.763.732.329 + 4.074.704.605 \bmod 4.294.967.296$

$= 5.838.436.934 \bmod 4.294.967.296$

$= 1.543.469.638$

$D = 1.769.234.797 + 2.915.898.378 \bmod 4.294.967.296$

$= 4.685.133.175 \bmod 4.294.967.296$

$= 390.165.879$

2. Iterasi

Iterasi dilakukan sebanyak 20 kali. Setiap kali iterasi mengikuti aturan sebagai berikut :

```
t ← ROTL ((X[1]*(2*X[1]+1)), 5)
u ← ROTL ((X[3]*(2*X[3]+1)), 5)
X[0] ← (ROTL((X[0] XOR t), u )) + S[2*i]
X[2] ← (ROTL((X[2] XOR u), t )) + S[2*i + 1]
Temp ← X[0]
X[0] ← X[1]
X[1] ← X[2]
X[2] ← X[3]
X[3] ← Temp
```

Nilai t dan u didapat dari blok B dan D diproses dengan fungsi f(x) = x(2x+1), kemudian dilanjutkan dengan menggeser nilai t dan u ke kiri sejauh 5 bit.

t = (B * (2 * B + 1)) mod 2^{32}

= (1.543.469.638 * (2 * 1.543.469.638 + 1)) mod 2^{32}

= (1.543.469.638 * (3.086.939.276 + 1)) mod 2^{32}

= (1.543.469.638 * 3.086.939.277) mod 2^{32}

= 4.764.597.048.399.171.726 mod 4.294.967.296

= 2.276.790.414

t : (dalam biner) 10000111101101010001000010001110

t : (digeser 5 bit) 11110110101000100001000111010000

t : (dalam desimal) 4.137.816.528.

Nilai 5 bit terakhir dari t yaitu 10000, atau dalam desimal sebesar 16, akan dipergunakan sebagai nilai penggeser blok C pada proses berikutnya, sejauh 16 bit.

u = (D * (2 * D + 1)) mod 2^{32}

= (390.165.879 * (2 * 390.165.879 + 1)) mod 2^{32}

= (390.165.879 * (780.331.758 + 1)) mod 2^{32}

= (390.165.879 * 780.331.759) mod 2^{32}

= 304.458.826.661.851.161 mod 4.294.967.296

= 2.481.549.337

u : (dalam biner) 10010011111010010111000000011001

u : (digeser 5 bit) 01111101001011100000001100110010

u : (dalam desimal) 2.100.167.474.

Nilai 5 bit terakhir dari u yaitu 10010, atau dalam desimal sebesar 18, akan dipergunakan sebagai penggeser blok A pada proses berikutnya, sejauh 18 bit.

Maka didapatkan nilai-nilai sebagai berikut :

- t = 4.137.816.528
- u = 2.100.167.474
- penggeser t = 16
- penggeser u = 18

Langkah selanjutnya adalah memproses blok A dan C dengan nilai-nilai yang telah dihasilkan.

A = (ROTL((A XOR t), u)) + S[2*i]

A : 1.852.532.084, dalam biner 01101110011010110110010101110100

t : 4.137.816.528, dalam biner 11110110101000100001000111010000 ⊕

A : (hasil xor) 10011000110010010111010010100100

A : (digeser 18 bit) 11010010100100100110001100100101

A : (dalam desimal) 3.532.808.997

Nilai A dijumlahkan dengan sub kunci S(2), dalam modulo 2^{32} :

A = 3.532.808.997 + 2.647.187.362 modulo 2^{32}

= 6.179.996.359 mod 4.294.967.296

= 1.885.029.063

C = (ROTL((C XOR u), t)) + S[2*i + 1]

C : 1.919.903.342, dalam biner 01110010011011110110011001101110

u : 2.100.167.474, dalam biner 01111101001011100000001100110010 ⊕

C : (hasil xor) 00001111010000010110010101011100

C : (digeser 16 bit) 01100101010111000000111101000001

C : (dalam desimal) 1.700.532.033

Nilai C dijumlahkan dengan sub kunci S(3), dalam modulo 2^{32} :

C = 1.700.532.033 + 426.203.967 modulo 2^{32}

= 2.126.736.000 mod 4.294.967.296

= 2.126.736.000

Maka didapat nilai masing-masing blok adalah :

A : 1.885.029.063

B : 1.543.469.638

C : 2.126.736.000

D : 390.165.879

Langkah berikutnya adalah mempertukarkan nilai blok dengan aturan (A, B, C, D) ← (B, C, D, A), sehingga pada iterasi pertama, didapat nilai pada masing-masing blok sebagai berikut :

A : 1.543.469.638

B : 2.126.736.000

C : 390.165.879

D : 1.885.029.063

Nilai masing-masing blok akan dilanjutkan pada iterasi berikutnya, dengan perhitungannya dapat dilihat pada Lampiran C.

4.4 Perancangan

Pada subbab ini akan diuraikan tentang perancangan basis data, perancangan modul perangkat lunak dan perancangan antarmuka.

4.4.1 Perancangan Basis Data

Pada tahap ini dilakukan perancangan terhadap basis data pada sistem baru. Berdasarkan análisis basis data pada sistem baru pada subbab 4.2.2, maka

dilakukan perubahan pada tabel-tabel yang dienkrip, yaitu tabel_bku, tabel_buku_bank dan tabel_akses. Tipe data pada tabel-tabel tersebut diubah menjadi tipe data *nvarchar* dan struktur tabel_bku diubah menurut analisa basis data. Perancangan tabel dapat dilihat pada Tabel 4.2.

Tabel 4.25 Perancangan Tabel

No	Nama Tabel	Nama Kolom	Tipe Data	Nulls
1	tabel_akses	nama_akses	nvarchar(50)	Not Nulls
		nama_pengguna	nvarchar(50)	Not Nulls
		kata_kunci	nvarchar(50)	Not Nulls
		hak_akses	nvarchar(50)	Not Nulls
		status	nvarchar(50)	Not Nulls
2	tabel_bku	no_bku	nvarchar(50)	Not Nulls
		tanggal_transaksi	nvarchar(50)	Not Nulls
		uraian_belanja	nvarchar(50)	Not Nulls
		uraian	nvarchar(1000)	Not Nulls
		jumlah_transaksi	nvarchar(50)	Not Nulls
3	tabel_buku_bank	tanggal_transaksi	nvarchar(50)	Not Nulls
		uraian	nvarchar(1000)	Not Nulls
		penerimaan	nvarchar(50)	Not Nulls
		pengeluaran	nvarchar(50)	Not Nulls

4.4.2 Perancangan Modul Perangkat Lunak

Modul yang akan ditambahkan pada perangkat lunak yaitu :

1. Modul Enkripsi

   Modul ini berisi fungsi-fungsi yang diperlukan untuk mengubah data transaksi yang dihasilkan aplikasi menjadi chiperteks, yang kemudian akan disimpan kedalam basis data. Algoritma enkripsi RC6 diimplementasikan pada modul ini.

2. Modul deskripsi

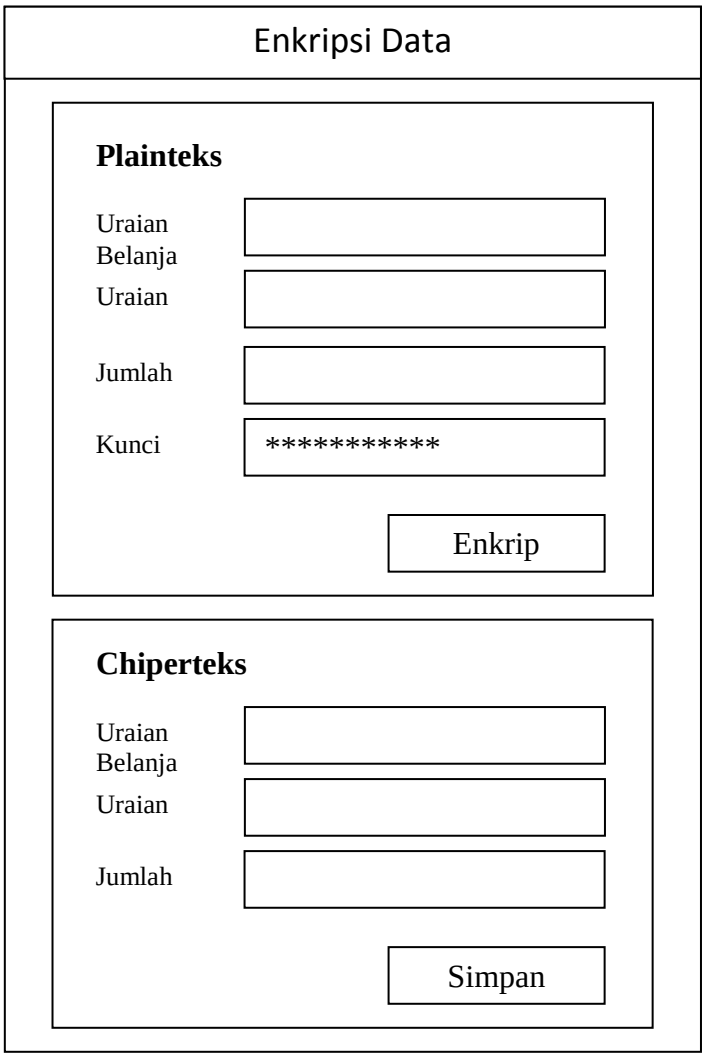
   Modul ini berisi fungsi deskripsi algoritma RC6 yang berguna untuk melakukan deskripsi.

4.4.3 Perancangan Antarmuka

Berikut ini dilakukan perancangan antarmuka dari fungsi tambahan yang diterapkan pada aplikasi keuangan, yaitu :

1. Perancangan Antarmuka Enkripsi Data

Antarmuka ini akan ditampilkan ketika pengguna melakukan proses penyimpanan data kedalam basis data. Pada tampilan ini akan diperlihatkan plainteks yang akan disimpan dan chiperteks yang dihasilkan setelah proses enkripsi. Gambar hasil perancangan antarmuka untuk enkripsi data dapat dilihat pada Gambar 4.14



Gambar 4.14 Perancangan Antarmuka Enkripsi Data

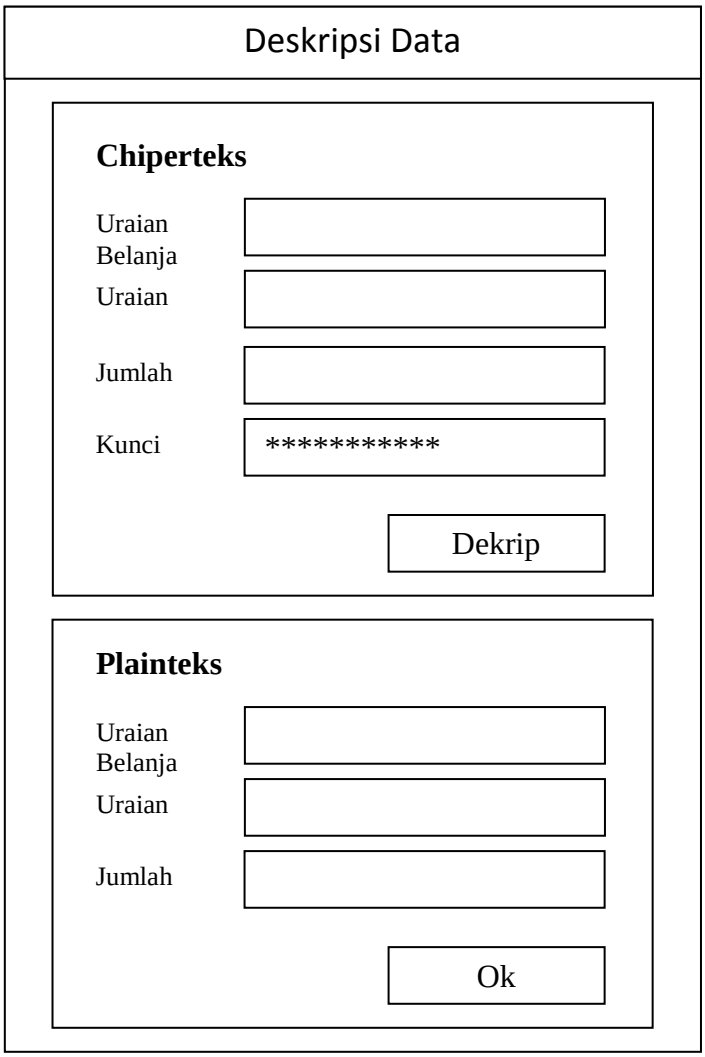
Keterangan gambar :

1. Uraian belanja, uraian dan jumlah pada bagian plainteks adalah *field* yang menjadi masukan untuk proses enkripsi.
2. Kunci adalah kunci masukan pengguna yang menjadi masukan untuk pembangkitan sub kunci yang digunakan pada proses enkripsi.
3. Enkrip adalah tombol perintah untuk melakukan proses enkripsi.

4. Uraian belanja, uraian dan jumlah pada bagian chiperteks adalah *field* yang memperlihatkan hasil dari proses enkripsi dari uraian belanja, uraian dan jumlah pada bagian plainteks, yang sudah berupa chiperteks.
5. Simpan adalah tombol perintah untuk menyimpan hasil dari enkripsi.

2. Perancangan Antarmuka Deskripsi Data

Antarmuka ini akan ditampilkan ketika pengguna melakukan proses untuk melihat data transaksi yang tersimpan di dalam basis data. Pada tampilan ini akan diperlihatkan chiperteks yang tersimpan dan dikembalikan ke bentuk plainteks setelah proses deskripsi. Gambar hasil perancangan antarmuka untuk deskripsi data dapat dilihat pada Gambar 4.15



Gambar 4.15 Perancangan Antarmuka Deskripsi Data

Keterangan gambar :

1. Uraian belanja, uraian dan jumlah pada bagian chiperteks adalah *field* yang menjadi masukan untuk proses deskripsi, yang berupa chiperteks.
2. Kunci adalah kunci masukan pengguna yang menjadi masukan untuk pembangkitan sub kunci yang digunakan pada proses deskripsi.
3. Dekrip adalah tombol perintah untuk melakukan proses deskripsi.

4. Uraian belanja, uraian dan jumlah pada bagian plainteks adalah *field* yang memperlihatkan hasil dari proses deskripsi dari uraian belanja, uraian dan jumlah pada bagian chiperteks, yang sudah berupa plainteks.

BAB V

IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dibahas mengenai implementasi dari hasil analisis dan perancangan perangkat lunak yang telah dilakukan pada bab sebelumnya, dan juga akan dijelaskan hasil yang didapatkan dari pengujian perangkat lunak yang telah dibangun.

5.1 Lingkungan Implementasi

Implementasi yang dilakukan menggunakan sebuah perangkat komputer, dengan spesifikasi sebagai berikut :

1. Prosesor Intel Pentium 4 1,5 GHz
2. Memory 512 MHz
3. Hardisk 80 GB
4. Perangkat Masukan Keyboard dan Tetikus
5. Perangkat Keluaran Monitor

Adapun perangkat lunak yang digunakan dalam melakukan implementasi adalah sebagai berikut :

1. Sistem Operasi Windows XP *Service Pack* 2
2. Basis data Microsoft SQL Server 2005 *Express*
3. Bahasa Pemrograman Microsoft Visual Basic 6.0

5.2 Batasan Implementasi

Perangkat lunak yang dibangun memiliki batasan sebagai berikut :

1. Fungsi enkripsi dan deskripsi diterapkan pada aplikasi keuangan hanya untuk melakukan pengamanan terhadap isi dari basis data keuangan.
2. Besar blok chiper yang dipergunakan adalah sebesar 128 bit, sehingga *word* sebesar 32 bit, jumlah putaran yang dipergunakan sebesar 20 kali putaran dan panjang kunci sebesar 16 *byte*.
3. Diasumsikan kunci pengguna disimpan diluar sistem.
4. Perangkat lunak tidak menangani fungsi untuk mengubah kunci yang dipergunakan untuk melakukan enkripsi dan deskripsi.

5.3 Implementasi Modul Perangkat Lunak

Pada tabel 5.1 dapat dilihat daftar hasil implementasi modul-modul yang telah dirancang.

Tabel 5.1 Implementasi Modul Perangkat Lunak

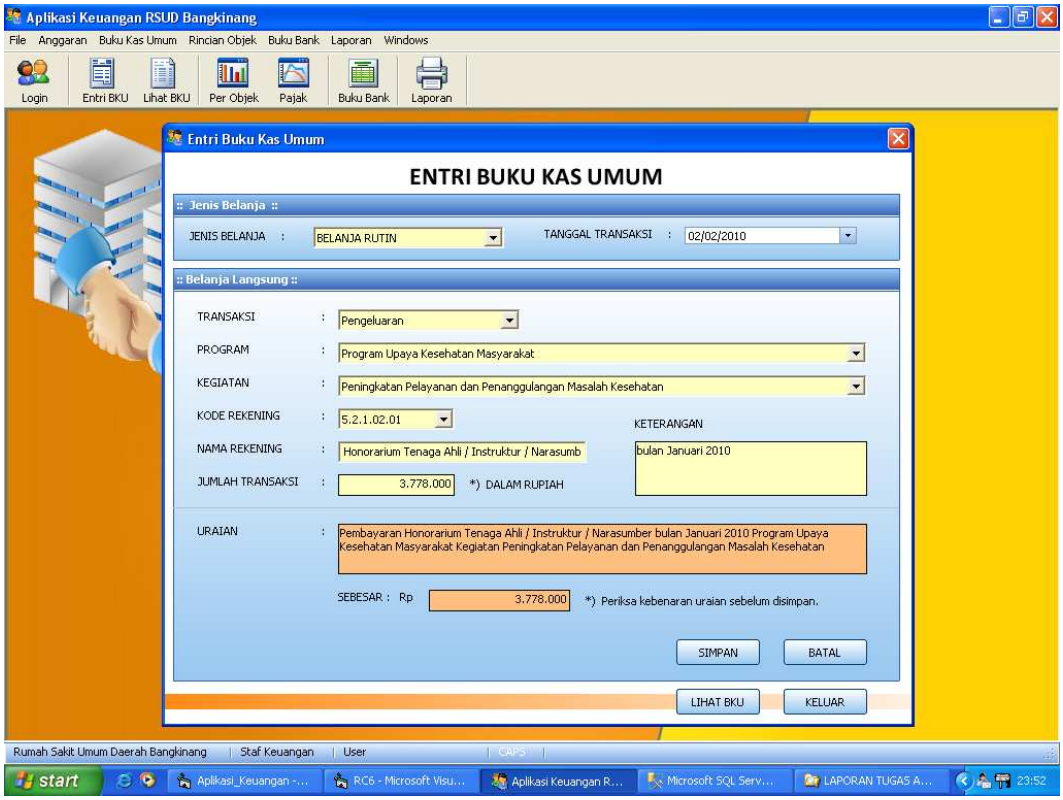
No	Nama Modul	Nama File yang Terlibat
1	Modul Enkripsi	enkrip_RC6.bas
2	Modul Deskripsi	dekrip_RC6.bas
3	Modul ROTL	hasil_ROTL.bas
4	Modul ROTR	hasil_ROTR.bas
5	Modul Desimal to Biner	DecToBin.bas
6	Modul Biner to Decimal	BinToDec.bas
7	Modul Mod	hasil_Mod.bas
8	Modul XOR	hasil_XOR.bas
9	Modul Fungsi	hasil_Fungsi.bas

5.4 Implementasi Antarmuka

Subbab ini berisi gambar hasil implementasi antarmuka dari perangkat lunak yang telah dibangun.

5.4.1 Antarmuka Masukan Data

Langkah pertama pada proses enkripsi adalah melakukan masukan data pada form Entri Buku Kas Umum, dengan tampilan dapat dilihat pada Gambar 5.1.



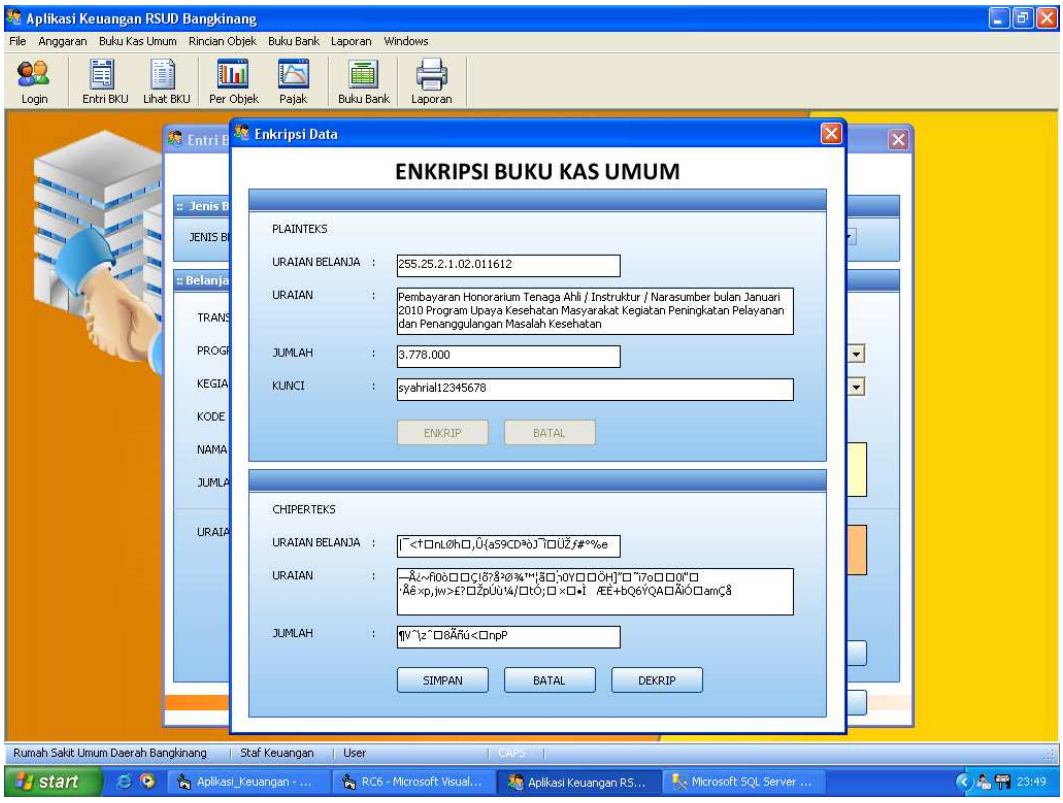
Gambar 5.1 Antarmuka Masukan Data

Setelah pengisian data selesai dilakukan, langkah berikutnya pengguna menekan tombol simpan, maka tampilan berikutnya akan menampilkan form enkripsi data.

5.4.2 Antarmuka Enkrip Data

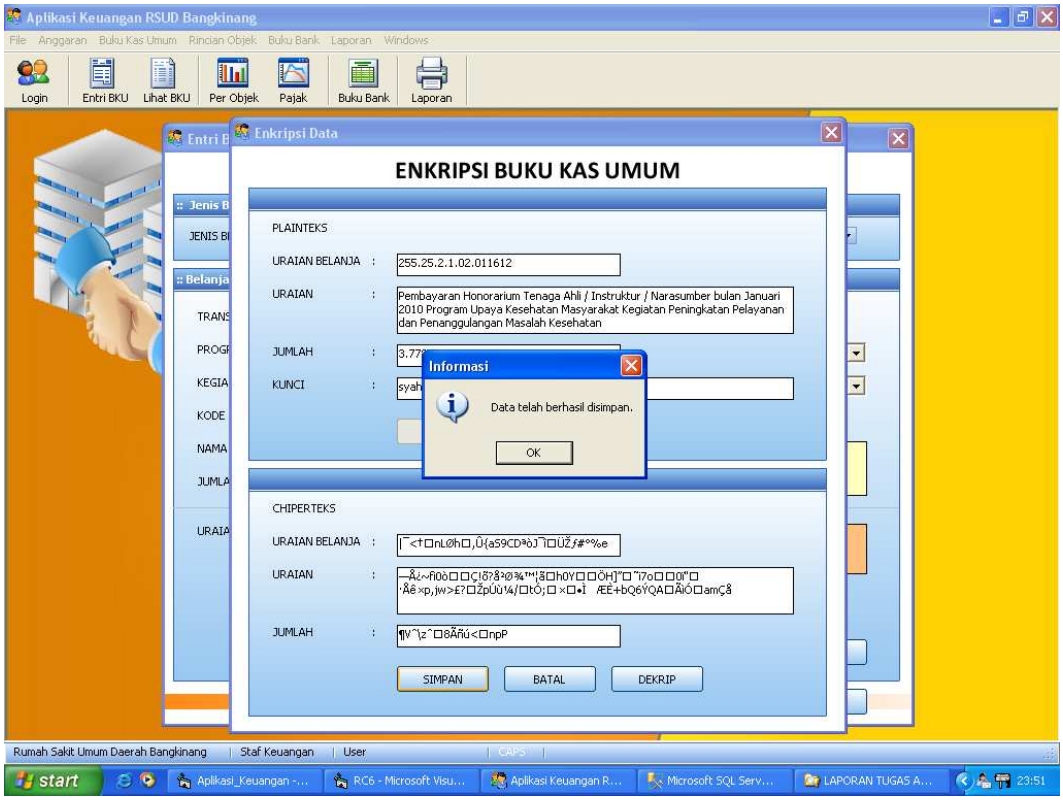
Form enkripsi data menampilkan data masukan yang diterima dari form masukan data. Untuk melakukan enkripsi data, pengguna harus memasukkan kunci enkripsi pada masukan kunci. Apabila pengguna tidak memasukkan kunci, maka akan ada peringatan bahwa kunci belum dimasukkan.

Kemudian pengguna menekan tombol enkrip, maka akan proses enkrip akan dilakukan, dan hasil dari proses enkrip data tersebut diperlihatkan pada bagian chiperteks. Antarmuka enkripsi data dapat dilihat pada Gambar 5.2.



Gambar 5.2 Antarmuka Enkrip Data

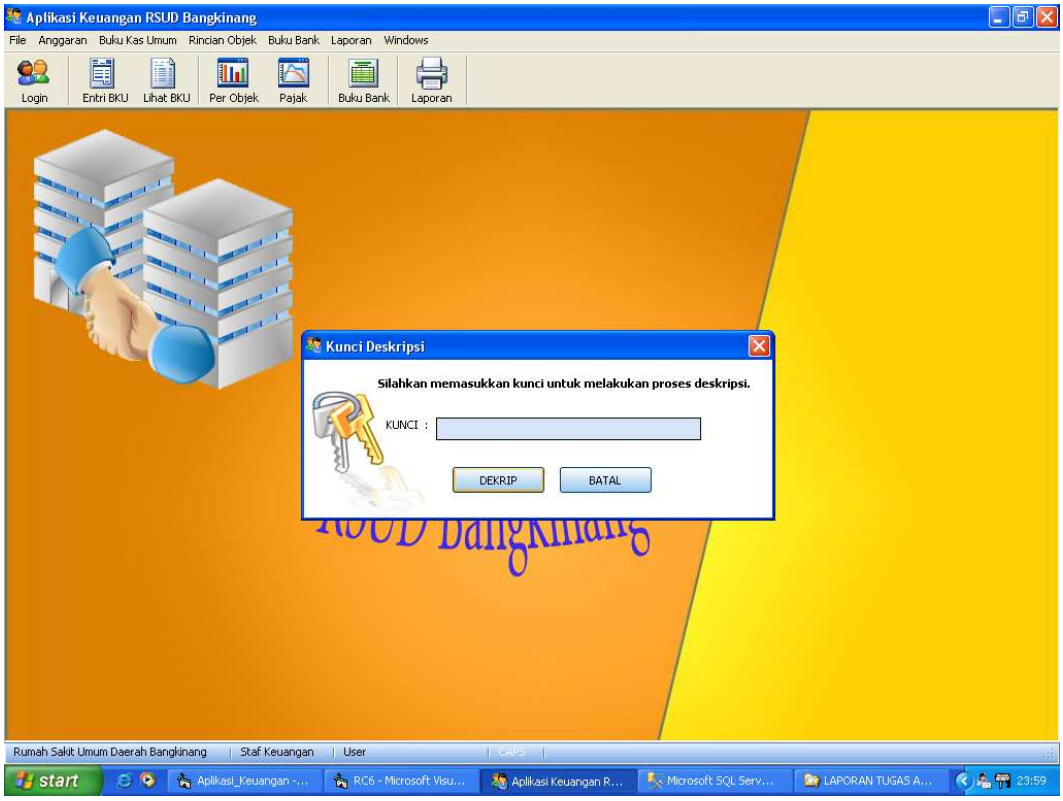
Kemudian bila pengguna menekan tombol simpan, maka data yang berbentuk chiperteks tersebut akan disimpan ke dalam basis data. Bila data telah berhasil disimpan, muncul infomasi bahwa data telah berhasil disimpan, seperti yang diperlihatkan pada Gambar 5.3.



Gambar 5.3 Informasi Penyimpanan Data

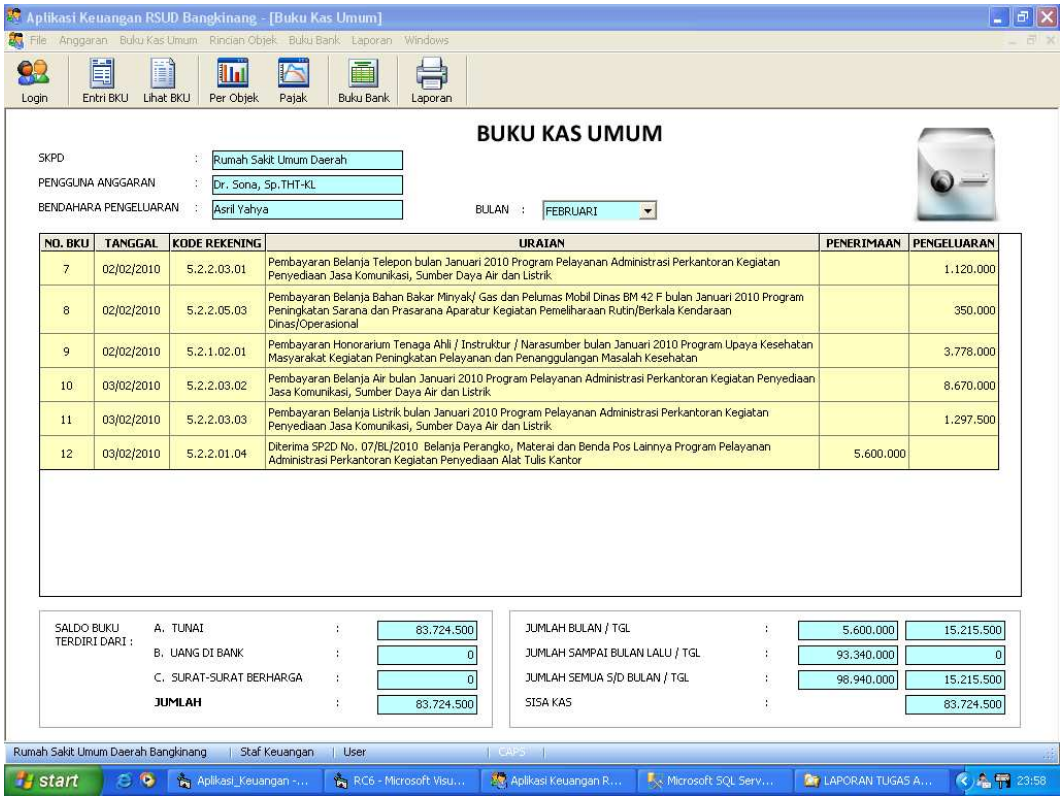
5.4.3 Antarmuka Dekrip Data

Proses dekrip data dimulai dari pengguna menekan sub menu Lihat BKU, kemudian muncul kotak dialog untuk memasukkan kunci dekrip, dengan antarmuka dapat dilihat pada Gambar 5.4. Pengguna memasukkan kunci dekrip dan menekan tombol dekrip. Bila pengguna belum memasukkan kunci, akan muncul pesan bahwa pengguna belum mengisi kunci.



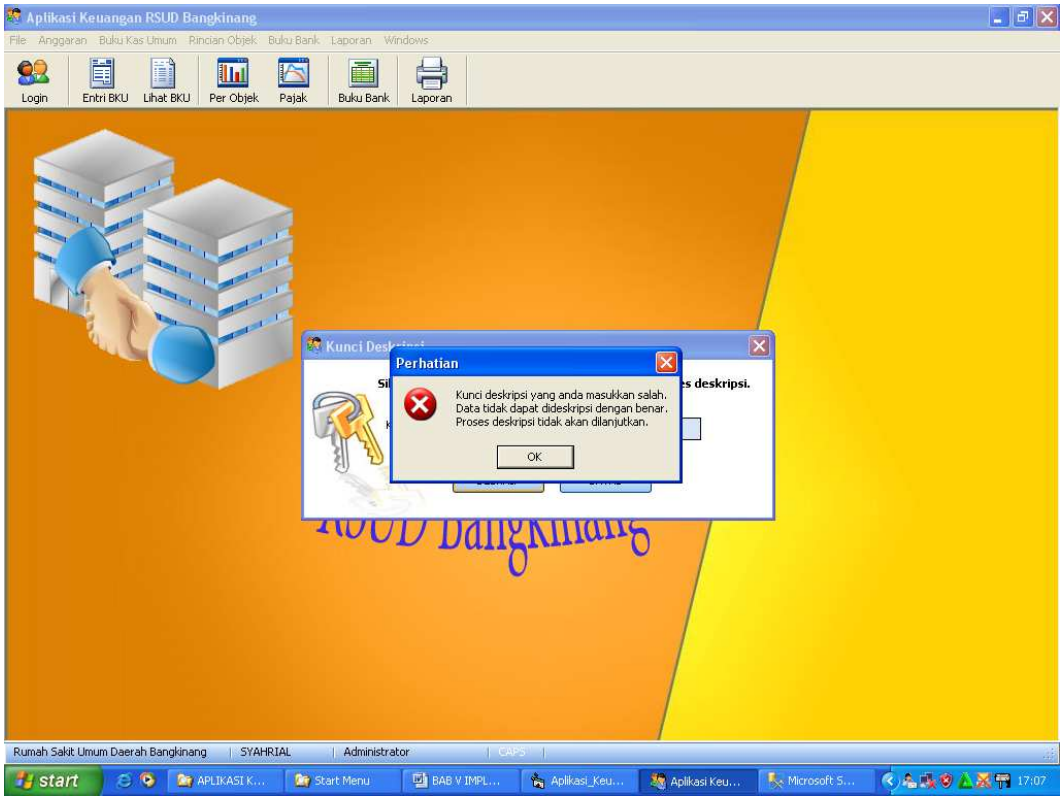
Gambar 5.4 Antarmuka Dekrip Data

Bila pengguna telah memasukkan kunci, maka proses deskripsi dilakukan. Bila pengguna memasukkan kunci yang benar, maka data yang terenkripsi dapat didekrip dengan baik. Data hasil deskripsi selanjutnya ditampilkan pada tampilan Lihat BKU, seperti yang diperlihatkan pada Gambar 5.5.



Gambar 5.5 Tampilan Lihat BKU

Apabila pengguna memasukkan kunci yang salah, maka data yang terenkripsi tidak dapat didekrip dengan benar, dan akan muncul pesan bahwa data tidak didekrip dengan benar, dan proses deskripsi tidak akan dilanjutkan. Tampilan dapat dilihat pada Gambar 5.6



Gambar 5.6 Pesan Kesalahan Dekrip Data

5.5 Pengujian

Pengujian yang dilakukan dibagi menjadi dua bagian yaitu pengujian enkripsi dan deskripsi, dan pengujian keamanan data terenkripsi.

5.5.1 Pengujian Enkripsi dan Deskripsi

Tujuan pengujian enkripsi dan deskripsi adalah untuk mengetahui apakah data telah dienkripsi dengan benar, dengan melakukan pengujian apakah hasil enkripsi dapat dideskripsi kembali dengan benar.

Skenario pengujian adalah dengan mengambil chiperteks dari basis data, dan melakukan deskripsi data tersebut dengan kunci enkripsi. Pengujian dilakukan dengan data dari tabel_bku dan tabel_buku_bank.

Adapun hasil dari 3 (tiga) kali percobaan enkripsi dan deskripsi dapat dilihat pada tabel 5.2. Asumsi dalam percobaan ini adalah menggunakan kunci sebesar 16 byte (128 bit), yaitu : syahrial12345678.

Tabel 5.2 Percobaan Enkripsi dan Deskripsi

No	Plainteks	Hasil Enkripsi	Hasil Deskripsi
1	Pembayaran Belanja Telepon bulan Januari 2010 Program Pelayanan Administrasi Perkantoran Kegiatan Penyediaan Jasa Komunikasi, Sumber Daya Air dan Listrik	□²»□èé}□- édk□'?ã'Ø¹&sŸí,□žæ à‰Uæ4¤□úq□□6Œ{_³□‹@® □qˆDô§ZeŽ]□IEçÝ¬’ó□®□Tþ ÏÈÞ>&Í□ëÔr□Ú□ÕEvÅÆ1²Ô¯ &j¼‘□½wÃéHn–C ¨ž□I 0	Pembayaran Belanja Telepon bulan Januari 2010 Program Pelayanan Administrasi Perkantoran Kegiatan Penyediaan Jasa Komunikasi, Sumber Daya Air dan Listrik

2	Pembayaran Belanja Bahan Bakar Minyak/ Gas dan Pelumas Mobil Dinas BM 42 F bulan Januari 2010 Program Peningkatan Sarana dan Prasarana Aparatur Kegiatan Pemeliharaan Rutin/Berkala Kendaraan Dinas/Operasional	□²» èé} édk '?ãÓÌYßEÑ·ïG _ƒ□v"'ª'[¢ Çïå ÐB uôÑ"Œ{>?"õKU¯ àœì}Ü-nBX Ìbð)O¬· ×X*ø÷¤ úq 6Œ{_³ ‹@® |.Ö†&3½™V–cß! »÷MzËûœwýs8^ !çðu º„óam v öTÔó□,µíie ;†Â©¶¨º¤f– Õ‰+¼VHüig £ 1Ñ°Ó7€‡u ¾F© µ5O¢õ´Ñû#JòÕY~¡W fpö,ÍØ	Pembayaran Belanja Bahan Bakar Minyak/ Gas dan Pelumas Mobil Dinas BM 42 F bulan Januari 2010 Program Peningkatan Sarana dan Prasarana Aparatur Kegiatan Pemeliharaan Rutin/Berkala Kendaraan Dinas/Operasional
3	Pembayaran Honorarium Tenaga Ahli / Instruktur / Narasumber bulan Januari 2010 Program Upaya Kesehatan Masyarakat Kegiatan Peningkatan Pelayanan dan Penanggulangan Masalah Kesehatan	—Å¿~fì0ò Ç!õ?å²Ø¾™¦ã h0Y ÖH]" ˜i7o 0ï" ·Åê×p,jw>£? ŽpÚù¼/ tÓ; × •Ì ÆÈ+bQ6Ý QA ÃìÓ□amÇå	Pembayaran Honorarium Tenaga Ahli / Instruktur / Narasumber bulan Januari 2010 Program Upaya Kesehatan Masyarakat Kegiatan Peningkatan Pelayanan dan Penanggulangan Masalah Kesehatan

5.5.2 Pengujian Keamanan Data Terenkripsi

Pengujian keamanan data terenkripsi dilakukan untuk mengetahui keamanan data yang telah disimpan dalam bentuk chiperteks terhadap serangan dari penyerang. Pengujian ini menggunakan jenis serangan *exhaustive attack* atau *brute force attack*. Percobaan yang dibuat untuk mengungkap plainteks atau kunci dengan mencoba semua kemungkinan kunci (*trial and error*).

Batasan pengujian keamanan data terenkripsi sebagai berikut :

1. Pengujian dalam proses *attack* terhadap data terenkripsi dilakukan sebanyak 25 kali percobaan.

Asumsi yang dipergunakan dalam pengujian terhadap *attack* :

1. Chiperteks yang diuji adalah uraian-uraian dari aplikasi keuangan.
2. Kriptanalisis memasukkan kemungkinan kunci yang digunakan secara acak.

Tabel 5.3 Pengujian Keamanan Data Terenkripsi terhadap Serangan

No	Kunci Awal	Kunci Uji Coba	Hasil
1	syahrial12345678	12345678AAAAAAAA	Tidak berhasil
2	syahrial12345678	12345678BBBBBBBB	Tidak berhasil
3	syahrial12345678	12345678CCCCCCCC	Tidak berhasil
4	syahrial12345678	12345678DDDDDDDD	Tidak berhasil
5	syahrial12345678	12345678EEEEEEEE	Tidak berhasil
6	syahrial12345678	12345678FFFFFFFF	Tidak berhasil
7	syahrial12345678	12345678GGGGGGGG	Tidak berhasil
8	syahrial12345678	12345678HHHHHHHH	Tidak berhasil
9	syahrial12345678	12345678IIIIIIII	Tidak berhasil
10	syahrial12345678	12345678JJJJJJJJ	Tidak berhasil
11	syahrial12345678	12345678KKKKKKK	Tidak berhasil
12	syahrial12345678	12345678LLLLLLLL	Tidak berhasil
13	syahrial12345678	12345678MMMMMMMM	Tidak berhasil
14	syahrial12345678	12345678NNNNNNNN	Tidak berhasil
15	syahrial12345678	12345678OOOOOOOO	Tidak berhasil
16	syahrial12345678	12345678PPPPPPPP	Tidak berhasil
17	syahrial12345678	12345678QQQQQQQQ	Tidak berhasil
18	syahrial12345678	12345678RRRRRRRR	Tidak berhasil
19	syahrial12345678	12345678SSSSSSSS	Tidak berhasil
20	syahrial12345678	12345678TTTTTTTT	Tidak berhasil
21	syahrial12345678	12345678syahrial	Tidak berhasil
22	syahrial12345678	12345678SYAHRIAL	Tidak berhasil

23	syahrial12345678	SYAHRIAL12345678	Tidak berhasil
24	syahrial12345678	1234567800000000	Tidak berhasil
25	syahrial12345678	00000000ZZZZZZZZ	Tidak berhasil

5.5.3 Kesimpulan Pengujian

Setelah melakukan pengujian terhadap enkripsi dan deskripsi data, dan pengujian keamanan data terenkripsi, dapat diambil kesimpulan sebagai berikut :

1. Implementasi RC6 telah dapat dilakukan, dimana data yang telah disimpan kedalam dalam basis data dalam bentuk chiperteks, dapat dideskripsi kembali menjadi plainteks atau teks semula.
2. Pengujian keamanan hasil enkripsi menggunakan metode *exhaustive attack* atau *brute force attack* menunjukkan hasil bahwa dari 25 kali usaha percobaan kunci dengan kunci yang salah, persentase kegagalan sebesar 100%, artinya data tidak dapat dibuka dengan menggunakan kunci yang berbeda dengan kunci saat melakukan enkripsi.

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan pengujian yang dilakukan pada bab Implementasi dan Pengujian, dapat diambil kesimpulan sebagai berikut :

1. Algoritma RC6 dapat diimplementasikan dengan baik untuk mengenkripsi basis data. Dari hasil pengujian dapat dilihat bahwa data didalam basis data keuangan dapat dienkripsi dan dideskripsi kembali dengan benar, sehingga data dapat diproses pada level aplikasi dengan baik serta tidak mengganggu struktur basis data.
2. Algoritma dapat melakukan pengamanan terhadap basis data keuangan, dimana dengan percobaan serangan dengan menggunakan metode *exhaustive attack* terhadap kunci pengguna sebanyak 25 kali percobaan, data tidak dapat ditembus dengan menggunakan kunci yang salah.

6.2 Saran

Beberapa hal sebagai saran untuk kemungkinan pengembangan sistem pada masa yang akan datang adalah sebagai berikut :

1. Pada aplikasi ini, terdapat kelemahan yaitu belum terdapat fungsi untuk mengubah kunci enkripsi. Untuk pengembangan sistem, dapat ditambahkan fungsi tersebut, sehingga kunci pengguna tidak bersifat statis. Misalnya

dikhawatirkan kunci enkripsi diketahui pihak yang tidak berhak, maka kunci enkripsi tersebut dapat diubah.

2. Panjang kunci pengguna pada aplikasi ini dibatasi maksimal 16 *byte*. Untuk pengembangan sistem, dapat dengan memberikan panjang kunci yang bervariasi, tidak terbatas hanya sepanjang 16 *byte*.

DAFTAR PUSTAKA

Ekklesia, Dicky "*Studi dan Implementasi Pengamanan Basis Data dengan Teknik Kriptografi Stream Chiper"*, [Online] Available http://www.informatika.org/~rinaldi/TA/Makalah_TA%20Dicky.pdf, diakses 2 Desember 2008.

Fauzan, Muhammad Firda, *"Pengamanan Transmisi dan Data Query Basis Data dengan Algoritma Kriptografi RC4"*, Halaman II-1, Bandung, Institut Teknologi Bandung, 2008.

Hapsari, Chitra dan Anisa Herdiani dan Ulya Raniasti "*Desain Implementasi Teknik Kriptografi untuk Pengamanan Basis Data Perusahaan*", [Online] Available http://www.informatika.org/~rinaldi/Kriptografi/2005-2006/Makalah/Makalah *2005-01.pdf, diakses 2 Desember 2008.*Meyer, Carl, dan Matyas, dan Sthepen M, "*Cryptography : A New Dimension in Computer Data Security*", NewYork , John Wiley & Sons, 1982.

Munir, Rinaldi. *"Kriptografi"*. Halaman 5-6, 9, 13-15, 116-134, Bandung : Penerbit Informatika, 2006.

Permana, Rangga Wisnu Adi, "*Implementasi Algoritma RC6 untuk Enkripsi SMS pada Telepon Selular*", Halaman II-4 – II-15, Bandung : Institut Teknologi Bandung, 2008.

Rivest , R.L, dan M.J.B. Robshaw, R. Sydney, Y.L. Yin. "*The RC6 TM Block Chiper*" Halaman 2-5, Paper, 1998.

Rudianto "*Analisis Keamanan Algoritma Kriptografi RC6*", [Online] Available http://www.informatika.org/~rinaldi/Kriptografi/2007-2008/Makalah1/Makalah IF5054-2007-A-051.pdf, diakses 2 Desember 2008.

Safrina, Rika. "*Studi dan Perbandingan Sistem Penyandian Pesan dengan Algoritma RC2, RC4, RC5 dan RC6*", [Online] Available http://www.informatika.org/~rinaldi/Kriptografi/2006-2007/Makalah1/Makalah1-079.pdf, diakses 2 Desember 2008

Silberschatz, A., H.F. Korth, Dan S. Sudarshan. "*Database System Concepts"*, McGraw-Hill, 4th Edition, 2002